

IMAGE RECONSTRUCTION ALGORITHMS
FOR ACHIEVING HIGH-RESOLUTION
POSITRON EMISSION TOMOGRAPHY IMAGES

By

JI-HO CHANG

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
UNIVERSITY OF FLORIDA

2004

Copyright 2004

by

Ji-Ho Chang

I dedicate this work to my family.

ACKNOWLEDGMENTS

First of all, I would like to thank my academic advisor, Dr. John M.M. Anderson, for his great guidance of my research. I also would like to thank my committee members Dr. Jian Li, Dr. Fred J. Taylor, and Dr. Bernard A. Mair for their precious comments and corrections on my dissertation. Especially, I would like to thank Dr. Bernard A. Mair for his comments and corrections on convergence proof of the proposed algorithms. I thank Yoonchul Kim and Kerkil Choi who have shared with me valuable discussions related to my research. Finally, I am grateful to my family, who have always prayed for my health.

TABLE OF CONTENTS

	<u>page</u>
ACKNOWLEDGMENTS	iv
ABSTRACT	vii
CHAPTER	
1 BACKGROUND	1
1.1 Overview of Positron Emission Tomography (PET)	2
1.2 PET Scanner	3
1.3 Sources of Error	6
1.4 System Model for Emission Data	10
2 LITERATURE REVIEW	16
2.1 Penalized Maximum Likelihood Algorithms	17
2.2 Scatter Correction Methods	19
2.3 Summary of the Proposed Algorithms	20
3 PENALIZED MAXIMUM LIKELIHOOD ALGORITHM	23
3.1 Penalized Maximum Likelihood (PML) Algorithm	24
3.2 Convergence Proof	33
3.3 Properties of the PML Algorithm	37
4 ACCELERATED PENALIZED MAXIMUM LIKELIHOOD ALGORITHM	39
4.1 Convergence Proof	40
4.2 Accelerated Penalized Maximum Likelihood (APML) Algorithm .	44
4.3 Direction Vectors	52
4.4 Properties of the APML Algorithm	55
5 QUADRATIC EDGE PRESERVING ALGORITHM	57
6 JOINT EMISSION MEAN AND PROBABILITY MATRIX ESTIMATION	63
6.1 Scatter Matrix Model	65
6.2 Joint Minimum Kullback-Leibler distance Method	68
6.3 Probability Correction in Projection Space (PCiPS) Algorithm .	69

7	SIMULATIONS AND EXPERIMENTAL STUDY	79
7.1	Regularized Image Reconstruction Algorithms	80
7.1.1	Synthetic Data	82
7.1.2	Real Data	87
7.2	APML algorithm	112
7.3	PCIPS Algorithm	116
8	CONCLUSIONS AND FUTURE WORK	138
8.1	Conclusions	138
8.2	Future Work	142
APPENDIX		
A	HUBER'S SURROGATE FUNCTIONS	144
B	SURROGATE FUNCTIONS FOR PENALTY FUNCTION	145
C	STRICT CONVEXITY OF PML OBJECTIVE FUNCTION	147
D	SOLUTION TO UNCONSTRAINED OPTIMIZATION PROBLEM IN MODIFIED APML LINE SEARCH	149
E	CONVEXITY OF SURROGATE FUNCTIONS FOR OBJECTIVE FUNCTIONS IN APML LINE SEARCH	151
REFERENCES		154
BIOGRAPHICAL SKETCH		160

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

IMAGE RECONSTRUCTION ALGORITHMS
FOR ACHIEVING HIGH-RESOLUTION
POSITRON EMISSION TOMOGRAPHY IMAGES

By

Ji-Ho Chang

August 2004

Chair: John M. M. Anderson

Major Department: Electrical and Computer Engineering

In this dissertation, we present four algorithms for reconstructing high-resolution images in PET. The first algorithm, referred to as the penalized maximum likelihood (PML) algorithm, iteratively minimizes a PML objective function. At each iteration, the PML algorithm generates a function, called a surrogate function, that satisfies certain conditions. The next iterate is defined to be the nonnegative minimizer of the surrogate function. The PML algorithm utilizes standard de-coupled surrogate functions for the maximum likelihood objective function of the data and de-coupled surrogate functions for a certain class of penalty functions. As desired, the PML algorithm guarantees nonnegative estimates and monotonically decreases the PML objective function with increasing iterations. For the case where the PML objective function is strictly convex, which is true for the class of penalty functions under consideration, the PML algorithm has been shown to converge to the minimizer of the PML objective function.

The drawback of the PML algorithm is that it converges slowly. Thus, a “fast” version of the PML algorithm, referred to as the accelerated PML (APML) algorithm,

was developed where an additional search step, called a pattern search step, is performed after each standard PML iteration. In the pattern search step, an accelerated iterate, which has lower cost than the standard PML iterate, is found by solving a certain constrained optimization problem that arises at each pattern search step. The APML algorithm retains the nice properties of the PML algorithm.

The third algorithm, referred to as the quadratic edge preserving (QEP) algorithm, aims to preserve edges in the reconstructed images so that fine details, such as small tumors, are more resolvable. The QEP algorithm is based on an iteration dependent, de-coupled penalty function that introduces smoothing while preserving edges. The penalty function was developed by modifying the surrogate functions of the penalty function for the PML method.

In PET, there are several errors that have the net effect of introducing blur into the reconstructed images. We propose a method that aims to reduce blur in PET images. The method is based on the assumption that the “true” probability matrix for the observed emission data is a product of an unknown nonnegative matrix, called a scatter matrix, and a “conventional” probability matrix. Under the suggested framework, the problem is to jointly estimate the scatter matrix and emission means. We propose an alternating minimization algorithm to estimate them by minimizing a certain distance.

The algorithms are qualitatively and quantitatively assessed using synthetic phantom data and real phantom data.

CHAPTER 1 BACKGROUND

Medical imaging modalities, such as X-ray computed tomography and magnetic resonance imaging, are used to obtain images of anatomical structures within the human body. However, in certain medical applications, it is also important to get physiological information. The reason is because physiological changes can indicate disease states earlier than anatomical changes [1]. Positron emission tomography (PET) and single-photon emission computed tomography (SPECT) are widely used medical imaging modalities that acquire physiological information on both human and animal subjects.

In SPECT, physiological information is obtained by imaging the distribution of gamma-ray or X-ray emitting radio-isotopes within the human body [1]. After the radio-isotopes are introduced into the human body, the radio-isotopes decay and emit gamma-ray or X-ray single photons. A SPECT scanner detects these photons with help of collimators that are made of lead. Since a large percentage photons are absorbed by the lead collimators, the sensitivity and accuracy of SPECT is limited.

In PET, there is no need for lead collimators because collimation is performed by electronic circuits that are connected to the detectors. Consequently, PET possesses relatively high sensitivity and accuracy, when compared to SPECT. Although cost is the major limitation of PET, recent research advances, such as a less expensive materials for detectors and scanner configurations that need a smaller number of detectors than conventional scanners, have helped decrease its cost [1]. The main advantage of SPECT over PET is its substantially lower cost.

We present a brief overview of PET, PET scanner, and certain errors in PET in Section 1.1, 1.2, and 1.3, respectively. Then, we provide some background on the image reconstruction problem in PET.

1.1 Overview of Positron Emission Tomography (PET)

Most clinical applications of PET are in oncological cases involving the diagnosis and staging of cancer, treatment planning and monitoring of tumors, detection of recurrent tumors, and localization of biopsy sites in cases when there are tumors in the head or neck [2,3]. PET is also used in the diagnosis of coronary artery disease and other heart diseases [2,3].

In PET, physiological information is acquired by imaging the distribution of positron-emitting isotopes, such as ^{13}N , ^{15}O , ^{18}F , and ^{11}C , within the human body [3]. The positron-emitting isotopes are bound to compounds with known biochemical properties. Compounds that are labeled with positron-emitting isotopes are called radiopharmaceuticals. The choice of the radiopharmaceutical depends on its application. For example, 2-[^{18}F]fluoro-2-deoxy-d-glucose ($[^{18}\text{F}]$ FDG) is used for imaging brain tumors while $[^{13}\text{N}]$ ammonia is used for the detection of coronary artery disease [3]. After the radiopharmaceutical is introduced into the subject (injection or inhalation), positrons are emitted as the positron-emitting isotopes decay. An emitted positron annihilates with a nearby electron within the body causing the generation of two high energy (511keV) photons. The two photons, which can penetrate the subject, travel in almost opposite directions. Ideally, photon pairs that are generated by a positron annihilating with an electron will be detected by a pair of detectors.

If two electronically connected detectors detect a pair of photons within a short time interval (e.g., < 10 ns), then the detection is recorded along the line connecting the two detectors, which is called a line-of-response. In the absence of an error, the detection indicates that there is an annihilation somewhere along the line-of-response. The detection of a photon pair is referred to as a coincidence. In addition,

the timing interval of a scanner that is used to define a coincidence is called the scanner's coincidence timing window. Since there are many, many detector pairs in a PET scanner, sufficient information is available for reconstructing a map of the concentration of the radiopharmaceutical. For each detector pair, the number of coincidences that occur during the scan are summed. The emission data is the coincidence sums for all of the possible detector pairs.

The key idea in PET is that emission data depends on the distribution of the radiopharmaceutical within the subject being scanned, which in turn depends on the metabolism of the subject. Consequently, numerous researchers have developed algorithms that reconstruct PET images whose pixel values represent the distribution of the radiopharmaceutical, and, ultimately, the metabolism of the subject.

1.2 PET Scanner

Typical PET scanners have a diameter of 80 – 100 cm and an axial extent of 10 – 20 cm [4]. Figure 1–1 (a) shows a simplified PET scanner and Figures 1–1 (b) and (c) show two-dimensional views of the scanner. Usually, a PET scanner consists of hundreds of rectangular bundles of crystals that are formed to make between 20 – 30 rings of detectors, where each detector ring contains 300 – 600 detectors. Each bundle of crystals is connected to a few (e.g., 2 – 8) photo-multiplier tubes (PMTs). Figure 1–1 (d) shows such a block of crystals coupled to four PMTs. When a photon interacts with a crystal, light photons are emitted and the PMTs collect the photons. From the collected light, a PET scanner determines the crystal within which the scintillation occurs.

State-of-the-art PET scanners generally provide two scan modes: slice-collimated mode and fully three-dimensional mode [4]. In slice-collimated mode or septa-extended mode, thin tungsten rings, called septa, are placed between the detector rings. Figure 1–2 (a) illustrates slice-collimated mode in which coincidences are collected within a detector ring. In fully three-dimensional mode, the septa is removed from the

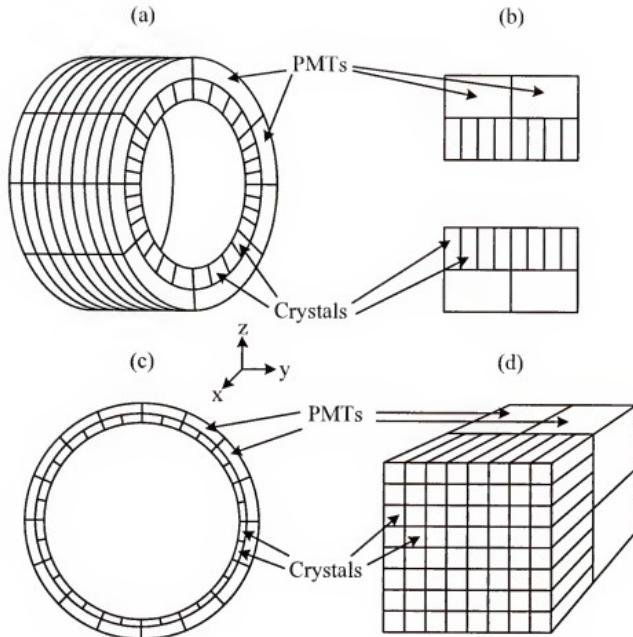


Figure 1.1: Simplified PET scanner: (a) a simplified full-ring PET scanner with 8 detector rings, (b) two-dimensional view of the scanner at $x = 0$, (c) two-dimensional view of the scanner at $y = 0$, and (d) a block detector consisting of an array of 8×8 crystals coupled to four PMTs, where the origin of the rectangular coordinate system is at the center of the scanner.

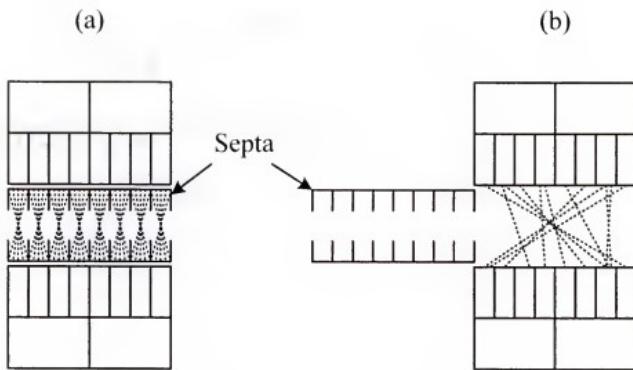


Figure 1-2: Scan modes: (a) septa-extended mode and (b) fully three-dimensional mode. The dotted lines represent line-of-responses.

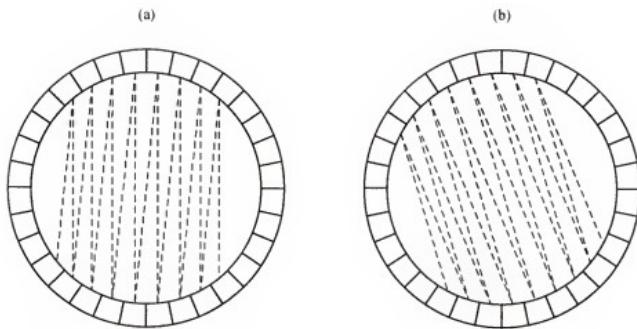


Figure 1-3: Projection definition using zig-zag scan: (a) a set of detector pairs that define a projection and (b) another projection. A dashed line represents a detector pair.

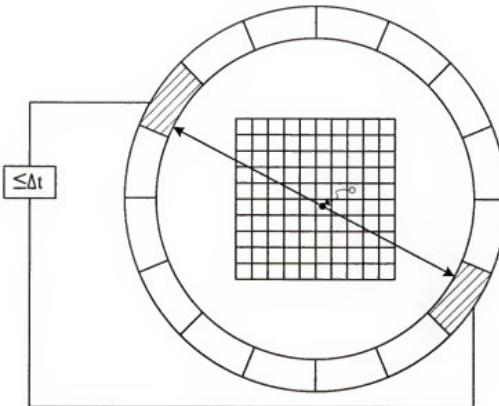


Figure 1-4: Illustration of positron range and a coincidence. Two concentric circles denote a detector ring.

scanner. Figure 1-2 (b) illustrates fully three-dimensional mode in which allowable line-of-responses are not restricted to occur within a detector ring.

Within a detector ring, detector pairs are grouped into projections, where a projection is a set of detector pairs that are defined by a particular “zig-zag scan”. Figure 1-3 (a) shows a projection and the defining zig-zag scan. Figure 1-3 (b) shows another projection. A sinogram is defined to be a $S \times T$ matrix, where each row of the matrix contains a projection, S is the number of detector pairs within the projection, and T is the number of projections in the emission data.

1.3 Sources of Error

After a positron is emitted, it travels a short distance before it annihilates with a nearby electron within a subject. The distance between the locations at which the emission and the annihilation take place is called positron range. The positron range is proportional to the reciprocal of the density of the material an emitted positron travels [5]. Figure 1-4 is an illustration of positron range for a simplified scanner geometry. The positron range depends on the energy an emitted positron deposits (and,

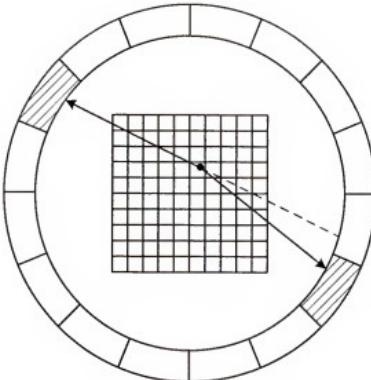


Figure 1-5: Illustration of non-collinearity of line-of-response. The dashed line indicates the path of the photon pair if they had departed in exactly opposite directions. The arrows show the actual photon paths.

consequently, the chosen positron-emitting isotope). For typical positron-emitting isotopes, a full width half maximum (FWHM) of the distribution of positron range is a few millimeters. For example, the FWHMs of the distribution of the positron range for ^{18}F and ^{15}O are about 2 mm and 8 mm, respectively [5, p. 331].

Two photons generated by an annihilation usually do not propagate in exactly opposite directions. This phenomenon is called non-collinearity of line-of-response. Figure 1-5 is an illustration of non-collinearity of line-of-response for a simplified scanner geometry. For a fixed direction that one of the two photons propagates, we refer to the angle between the opposite direction of the fixed direction and the actual direction that the other photon travels in as the “angle of non-collinearity”. The distribution of the angle of non-collinearity can be approximated by a Gaussian distribution with a FWHM of 0.5 degree [5].

The angle between the path that a photon propagates and the face of a detector when the photon hits the detector is referred to as the incident angle. The way that a photon interacts with a detector depends on the incident angle. If the path that a

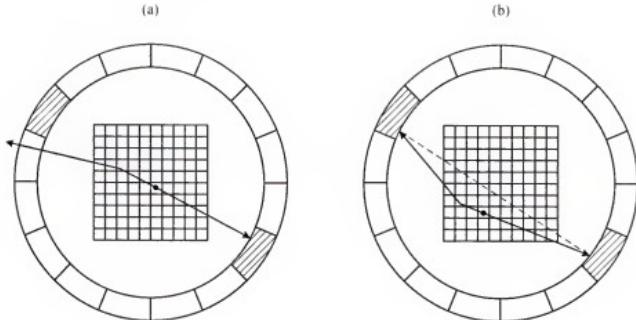


Figure 1-6: Single and scatter: (a) Illustration of attenuation and a single, and (b) Illustration of a scattered coincidence. The arrow penetrating the detector ring denotes that the photon is scattered through an oblique angle such that it does not hit a detector. The dotted line denotes the incorrectly positioned line-of-response.

photon travels is not perpendicular to the face of a detector when the photon hits the detector, then it is possible that the photon does not interact with the detector that it strikes. The photon may interact with another detector that is nearby the detector the photon hits originally. This phenomenon is termed by detector penetration [6, 7].

When a 511keV photon propagates within a subject and interacts with an electron, the photon may undergo a phenomenon known as Compton scattering. When a photon hits an electron, the photon gives part of its energy to the electron and deflects from its original path if the photon has sufficient energy. This phenomenon is called Compton scattering. Compton scattering can lead to three kinds of error: attenuation, scatter, and accidental coincidence.

Most scattered photons are scattered out of the scanner's field-of-view so that many of them are not detected. This phenomenon is called attenuation. Figure 1-6 (a) illustrates attenuation, where one of the two photons of an annihilation does not hit a detector due to Compton scattering. Figure 1-6 (a) also illustrates an event called a single, where only one of the emitted photons of an annihilation is

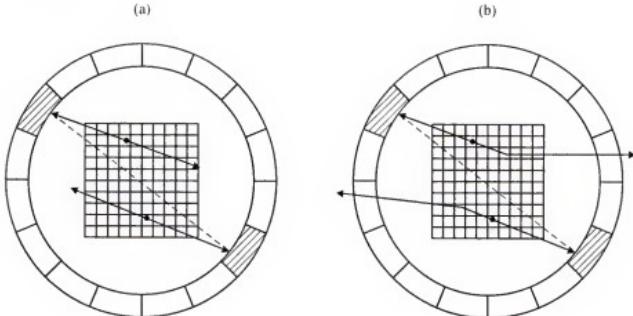


Figure 1-7: Accidental coincidence: (a) Illustration of an accidental coincidence due to two annihilations occurring at almost the same time and (b) Illustration of an accidental coincidence due to Compton scattering.

detected. Note that attenuation leads to an incorrect decrease in emission counts. In order to address attenuation, numerous correction methods have been proposed (see e.g., [8–10]).

Consider an annihilation event where Compton scattering occurs. It is still possible that a detector pair detects both photons even though one or both of the photons may have undergone Compton scattering. Such an event is called a scattered coincidence or scatter, which is illustrated in Figure 1-6 (b). Note that a scattered coincidence leads an incorrect increase in emission counts. Since a scattered photon loses part of its energy, the energy of detected photons may be used to discriminate between unscattered photons and scattered photons [11, pp. 65–69].

Two photons arising from different annihilations can be recorded by a detector pair. This event is called an accidental coincidence. Figure 1-7 (a) illustrates an accidental coincidence due to two annihilations occurring at almost the same time. Sometimes, an accidental coincidence may be due to Compton scattering as illustrated in Figure 1-7 (b). Like scatter, an accidental coincidence leads an incorrect increase

in emission counts. For many PET scanners, the mean accidental coincidence rate is estimated using a “delayed” timing window technique [12].

The efficiency of a detector pair is defined to be the probability that a coincidence is recorded when a photon pair hit the detectors. Ideally, this probability should be one. However, the efficiencies of detector pairs are non-uniform because of their geometric differences and the non-uniform physical characteristics of detectors. The non-uniformity of detector pairs is referred to as detector inefficiency. To address detector inefficiency, correction methods such as [13–15] have been proposed.

1.4 System Model for Emission Data

In 1982, Shepp and Vardi proposed a Poisson model for PET emission data and an algorithm, known as the maximum likelihood expectation maximization (MLEM) algorithm, for reconstructing maximum likelihood (ML) emission images [16]. In the Poisson model, the region-of-interest is divided into J equal sized volume elements, called voxels. Ultimately, the goal is to estimate the mean number of positrons emitted from each voxel. Let the i^{th} component of a vector \mathbf{d} , d_i , represents the observed number of photon pairs recorded by the i^{th} detector pair and the j^{th} component of a vector \mathbf{x} , x_j , represents the unknown mean number of emissions from the j^{th} voxel. Further, let I and J denote the number of detector pairs and number of voxels, respectively. The $I \times 1$ vector \mathbf{d} and the $J \times 1$ vector \mathbf{x} are the emission data and the unknown emission mean vector, respectively. Let \mathcal{P}_{ij} denotes the probability that an annihilation in the j^{th} voxel leads to a photon pair being recorded by the i^{th} detector pair. The $I \times J$ probability matrix \mathcal{P} has \mathcal{P}_{ij} as its $(i, j)^{\text{th}}$ element. In the Poisson model, Shepp and Vardi assumed that the emission data \mathbf{d} is an observation of a random vector \mathbf{D} with elements $\{D_i\}_{i=1}^I$ that are Poisson distributed and independent. For all i , the mean of the random variable D_i is

$$E\{D_i\} = \sum_{j=1}^J \mathcal{P}_{ij} x_j. \quad (1.1)$$

In practice, the probability is unknown and must be estimated somehow. The simplest way to estimate the probability matrix \mathcal{P} is the angle-of-view (AOV) method [16]. In the AOV method, a detector ring and a detector are modelled by a circle and an arc on the circle, respectively. Moreover, within a voxel, all emitted positrons are assumed to be emitted from the center of the voxel. Figure 1–8 illustrates a detector ring, a detector pair, and the tube (i.e., spatial extent) that is defined by the detector pair. In the figure, the AOV from the point g_j to the detector pair (y, z) is also shown. Specifically, this AOV is defined to be

$$\text{AOV from } g_j \text{ to } (y, z) \triangleq \begin{cases} \min\{\angle a_y g_j b_y, \angle a_z g_j b_z, (\pi - \angle a_y g_j b_z), (\pi - \angle b_y g_j a_z)\}, & g_j \in \text{tube } (y, z) \\ 0, & \text{otherwise} \end{cases} \quad (1.2)$$

Said another way, the AOV in (1.2) is the maximum angle for which a line that goes through the point g_j will simultaneously intersect both detectors of the detector pair (y, z) . In the AOV method, the probability \mathcal{P}_{ij} is defined to be

$$\mathcal{P}_{ij} \triangleq \frac{\text{AOV from } g_j \text{ to } (y, z)}{\pi}, \quad (1.3)$$

where the detector pair (y, z) is the i^{th} detector pair and the point g_j is the center point of the j^{th} voxel. In the AOV method, it is assumed that a photon is detected by a detector whenever the photon hits the arc corresponding to the detector. Clearly, the AOV method does not account for detector penetration discussed in Section 1.3. Some methods have been developed to address errors due to detector penetration [6,7].

In this dissertation, we make the following mild assumptions on the probability matrix \mathcal{P} and the emission data \mathbf{d} :

- **(AS1)** \mathcal{P} has no row vector of zeros
- **(AS2)** $\mathbf{d}' \mathcal{P}_j \neq 0$ for all j , where \mathbf{d}' is the transpose of \mathbf{d} and \mathcal{P}_j is the j^{th} column of \mathcal{P} .

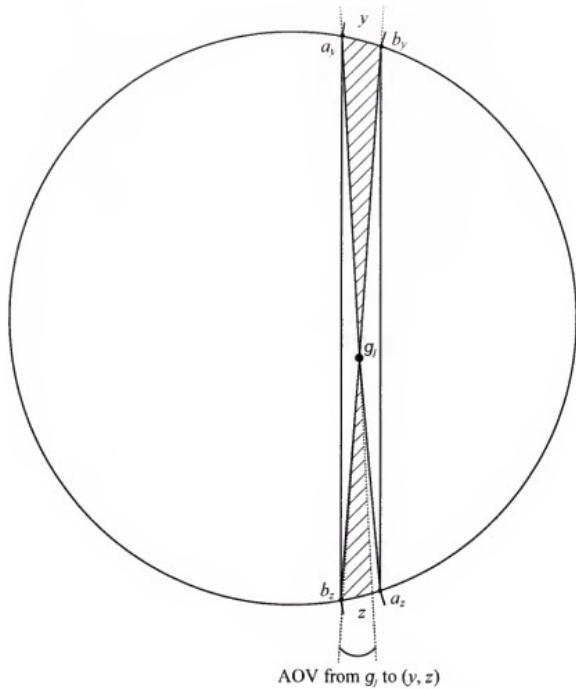


Figure 1-8: The angle-of-view from the point g_j to the detector pair (y, z) is shown. The circle denotes a detector ring. The arcs (a_y, b_y) and (a_z, b_z) represent the detectors y and z , respectively. The area between the vertical lines (a_y, b_z) and (b_y, a_z) represents the tube defined by the detector pair (y, z) .

To see the implication of the second assumption, consider all the detector pairs where the probability of recording a photon pair generated by an annihilation in the j^{th} voxel is non-zero. Among the set of detector pairs, the second assumption implies that there exists at least one detector pair with non-zero emission counts. The **(AS2)** is expected to hold whenever the duration of the emission scan is a reasonable length of time. In addition to **(AS1)** and **(AS2)**, we assume that the probability matrix \mathcal{P} accounts for errors due to attenuation, detector inefficiency, detector penetration, positron range, non-collinearity of line-of-response, and scatter. In practice, there are correction methods for attenuation [9], detector inefficiency [15], and detector penetration [7]. However, other correction methods for detector penetration, attenuation, and detector inefficiency could be used in conjunction. Note that in Chapter 6, we do not assume that the probability matrix accounts for errors due to scatter and non-collinearity of line-of-response. Instead, we present a method that estimates the probability matrix in such a way that errors due to scatter non-collinearity of line-of-response are addressed.

When accidental coincidences (i.e., randoms) are considered, the Poisson model must be modified so that now the emission data \mathbf{d} is an observation of a random vector \mathbf{D} that is Poisson distributed with mean $(\mathcal{P}\mathbf{x} + \boldsymbol{\rho})$, where the i^{th} component of $\boldsymbol{\rho}$, ρ_i , is the mean number of accidental coincidences recorded by the i^{th} detector pair, $i = 1, 2, \dots, I$. Usually, it is assumed that the mean accidental coincidence rate $\boldsymbol{\rho}$ is known. In practice, the mean accidental coincidence rate is estimated using a “delayed” time window technique [12].

Given the emission data \mathbf{d} , mean accidental coincidence rate $\boldsymbol{\rho}$, and probability matrix \mathcal{P} , the problem of interest is to estimate the mean number of positrons emitted from each voxel. Since it is assumed that the data are independent, it follows that

the likelihood function for emission data \mathbf{d} is given by

$$\Pr\{\mathbf{D} = \mathbf{d}|\mathbf{x}\} = \prod_{i=1}^I \frac{[\mathcal{P}\mathbf{x} + \boldsymbol{\rho}]_i^{d_i}}{d_i!} e^{-[\mathcal{P}\mathbf{x} + \boldsymbol{\rho}]_i}. \quad (1.4)$$

The ML estimate of \mathbf{x} is defined to be the maximizer of the likelihood function over the feasible set. Alternatively, the ML estimate of emission mean vector is given by

$$\mathbf{x}_{ML} \triangleq \arg \max_{\mathbf{x} \geq \mathbf{0}} \Psi(\mathbf{x}), \quad (1.5)$$

where $\Psi(\mathbf{x}) \triangleq \log \Pr\{\mathbf{D} = \mathbf{d}|\mathbf{x}\}$ is the log likelihood function:

$$\Psi(\mathbf{x}) = \sum_{i=1}^I d_i \log([\mathcal{P}\mathbf{x} + \boldsymbol{\rho}]_i) - \sum_{i=1}^I [\mathcal{P}\mathbf{x}]_i + \sum_{i=1}^I \{-\rho_i - \log(d_i!)\} \quad (1.6)$$

(note: maximizing the likelihood function or log likelihood function are equivalent operations).

Although the ML estimator has several nice theoretical properties [17, ch. 7], images produced by the ML method (i.e., the MLEM algorithm) have the drawback that they are extremely noisy. This is due to the PET image reconstruction problem is ill-posed because of the facts that (1) scan times for data acquisition is short, (2) emission data contain errors due to attenuation, scatters, and accidental coincidences, and (3) the data obey Poisson statistics. Currently, the most popular way to address the ill-posed nature of the image reconstruction problem is through the use of penalty functions. Numerous penalized maximum likelihood (PML) methods (also known as Bayesian and maximum *a posteriori* methods (MAP)) have been proposed [18–35].

In the MAP method, \mathbf{x} is assumed to be an observation of a random variable \mathbf{X} with known distribution and the *a posteriori* distribution (i.e., conditional probability density function of \mathbf{X} given $\mathbf{D} = \mathbf{d}$) is maximized. After some manipulations [17, p. 351], the MAP estimate is found to be the maximizer of the log likelihood function

plus the log of the probability density function of \mathbf{X} :

$$\mathbf{x}_{MAP} \stackrel{\Delta}{=} \arg \max_{\mathbf{x} \geq \mathbf{0}} \Psi(\mathbf{x}) + \log \Omega(\mathbf{x}), \quad (1.7)$$

where the function Ω is the probability density function of \mathbf{X} (i.e., prior distribution). It is through the prior distribution Ω that MAP methods have the ability to regularize the image reconstruction problem.

The form of the prior distributions commonly used in PET is $\Omega(\mathbf{x}) = C e^{-\beta \Lambda(\mathbf{x})}$, where the function Λ is a scalar valued function and C and β are constants. The constant $C > 0$ is chosen such that the area under the distribution Ω equals one. The constant $\beta > 0$, known as the penalty parameter, controls the penalty function's degree of influence. Since it is known that PET images should be highly correlated, the penalty function Λ is designed in such a way that it forces the estimates of neighboring voxels to be similar in value. Given the definition of Ω , the PML or MAP estimate is the nonnegative minimizer of the PML objective function

$$\Phi(\mathbf{x}) \stackrel{\Delta}{=} -\Psi(\mathbf{x}) + \beta \Lambda(\mathbf{x}). \quad (1.8)$$

CHAPTER 2

LITERATURE REVIEW

Although the MLEM algorithm by Shepp and Vardi [16] produces ML estimates of the emission means, the images produced by the ML method are extremely noisy due to the fact that the image reconstruction problem in PET is ill-posed. As discussed in Section 1.4, the reasons are that (1) scan times for data acquisition is short, (2) emission data contain errors due to attenuation, scatters, and accidental coincidences, and (3) the data obey Poisson statistics. One way to obtain PET images with sufficient smoothness is to terminate the MLEM algorithm before the log likelihood function is maximized. Of course, the resulting images are not ML images. Another modification of the MLEM algorithm is to first obtain an ML image and then filter it with a low pass filter. The drawback of this post-filtering is that it is not clear how the filter is chosen. A variation of filtering approach just described is to filter every MLEM iterate, which was suggested by Silverman [36]. Silverman did not provide an answer as to how the filter should be chosen. Denoising emission data (i.e., observed data) is another way to regularize the PET image reconstruction problem [37, 38].

Currently, the most popular way to introduce regularization is through the use of penalty functions that force the estimates of neighboring voxels to be similar in value. The basis for such penalty functions is that PET images should be highly correlated. The first part of the chapter is devoted to so-called penalized maximum likelihood (PML) algorithms. PML algorithms are algorithms that minimize PML objective functions, which are a sum of the negative log likelihood function and a penalty function. In PET, PML and maximum a posteriori (MAP) are terms that are used for methods that minimize PML objective functions. In Section 2.1, we briefly review existing PML algorithms.

Reconstructed PML images are blurred by errors due to detector penetration, positron range, non-collinearity, and scatter. Errors due to scatter are difficult to correct because scatter depends on the activity and attenuation within the subject and the scanner design. In Section 2.2, some scatter correction methods are reviewed. The regularized image reconstruction algorithms and scatter correction algorithm we propose are compared with the existing algorithms in Section 2.3.

2.1 Penalized Maximum Likelihood Algorithms

In 1990, Green [23] proposed a PML algorithm, known as the one-step-late (OSL) algorithm. The algorithm can be viewed as a fixed point iteration that is derived from the Kuhn-Tucker equations [39, pp. 36-49] for the PML optimization problem. Incidentally, Shepp and Vardi showed that the MLEM algorithm could be derived from the Kuhn-Tucker conditions in a similar way. The OSL algorithm is straightforward to implement, but nonnegative estimates cannot be guaranteed and, like many existing algorithms, convergence is an open issue. Lange's goal [24] was to modify the OSL algorithm in such a way that the modified algorithm converges to the PML estimate. It should be pointed out that Lange's algorithm requires line searches, which can be computationally expensive.

Alenius *et al.* [32] suggested a Gaussian "type" prior that depends on the median of voxels within local neighborhoods, and introduced an algorithm called the median-root-prior (MRP) algorithm. The MRP algorithm is based on an iteration dependent objective function. Consequently, it really cannot be considered a PML algorithm. Nevertheless, the MRP algorithm generates "good" images in the sense that noise level of the reconstructed images is suppressed. It should be mentioned that a PML algorithm was derived by Hsiao *et al.* [40] that resembles the MRP algorithm and performs similar to the MRP algorithm. The PML algorithm by Hsiao *et al.* was derived using a prior that is based on a certain auxiliary vector.

Levitin and Herman [21] proposed a PML algorithm based on an assumption that the prior distribution of the true emission means was a multivariate Gaussian distribution. The assumption led to a penalty function that was in the form of a weighted least-squares distance between \mathbf{x} and a reference image. However, they did not indicate how the reference image to be chosen.

An algorithm was proposed by Wu [27] using a wavelet decomposition formulation. Specifically, the author assumed that a vector consisting of the wavelet coefficients of the true emission means is a zero-mean Gaussian random vector with a known covariance matrix. From this assumption, a prior distribution for the emission means was derived. The prior distribution is a zero-mean Gaussian random vector with a covariance matrix that depends on the choice for the wavelet transform and the assumed distribution for the vector of wavelet coefficients. It should be pointed out that the assumption was not clearly justified in the paper.

Researchers have used an optimization algorithm, called the iterative coordinate descent (ICD) algorithm [41, pp. 283-287], to obtain estimates for various penalty functions [31], [42]. Convergence results are given for the penalized weighted least-squares method [42] and both algorithms (i.e., [31], [42]) enforce the nonnegativity constraint. Algorithms based on ICD algorithm update each voxel in a serial manner so that parallel implementation for them may not be possible.

De Pierro [25, 30] derives PML algorithms that minimize certain surrogate functions that he constructs by exploiting the fact that the log likelihood function is concave and penalty functions, such as the quadratic penalty function, are convex. Except for the quadratic penalty function, closed form expressions for the minimizers of the surrogate functions do not exist. Consequently, an optimization method, such as Newton's method [43, pp. 201-202], is needed to minimize the surrogate functions. De Pierro presents some convergence results, however the utility of his methods is unclear because no experimental results were provided. It should be noted that, in

the transmission tomography paper by Erdođan and Fessler [8], a quadratic surrogate function was used for a certain class of penalty functions. The quadratic surrogate function was developed by Huber [44, pp. 184-186].

A fast PML method, based on the ordered subset-EM algorithm [45], was proposed by De Pierro and Yamagishi [33]. The authors show that if the sequence generated by the algorithm converges, then it must converge to the true PML solution. Recently, Ahn and Fessler proposed algorithms [35] that are based on the ordered subset-EM algorithm [45], an algorithm by De Pierro and Yamagishi [33], and an algorithm by Erdođan and Fessler [10]. Like other algorithms based on the ordered subset-EM algorithm, there is some uncertainty as to how the subsets are to be chosen. In the paper by Ahn and Fessler, the algorithms are said to converge to the nonnegative minimizer of the PML objective function for certain penalty functions and their accompanying parameters by using a relaxation parameter that diminishes with iterations. Open issues are how the relaxation parameters should be chosen in practice and how they affect the performance of the algorithms. Convergence rate varies with the relaxation parameter.

2.2 Scatter Correction Methods

Many methods have been proposed to correct scattered coincidences [11, ch. 3]. They can be classified into a few categories: (1) energy window method, (2) convolution/deconvolution method, and (3) calculating scatter distribution method.

One of the scatter correction methods is based on the use of multiple (two or more) energy windows [46,47]. Recall that photons lose their energy when they have undergone Compton scattering. The principle of the method utilizing energy windows is to discard detection of a photon whenever the energy of the photon is less than 511Kev. Since detectors have finite energy resolution, there is a limitation of energy window based methods. Consequently, it is preferred to use another method jointly with the energy window techniques.

Another correction method for scatter is the convolution/deconvolution method [48–51]. The methods in [48, 49] assume that scattered coincidences (i.e., scatter) can be approximated by a convolution of unscattered coincidences and a certain scatter function. Under the assumption, the mean scatter in the observed coincidences is estimated that can be subtracted from emission data or incorporated in the system model for emission data. The method by McKee *et al.* assumes that the distribution of scattered annihilations can be approximated by the convolution of the distribution of unscattered annihilations and some scatter response function [50]. The issue of the convolution/deconvolution method is that the distribution of unscattered coincidences (or annihilations) and scatter response function (or scatter function) are not known.

Ollinger introduced a scatter correction method that calculates scatter distribution using an analytical equation, transmission images, emission images, and scanner geometry [52]. Computational cost of the method is excessively expensive, thus it might be hard to be accepted in clinical use at the moment of writing.

2.3 Summary of the Proposed Algorithms

- In Chapter 3, we present an algorithm that obtains PML estimates for a certain class of edge-preserving penalty functions. The PML algorithm is derived by combining the convexity idea by De Pierro [25, 30] and Huber’s surrogate functions [44]. Combining two existing theories in such a way that the PML algorithm is convergent is new to PET community. In theory, the algorithm guarantees nonnegative iterates, monotonically decreases the PML objective function with increasing iterations, and converges to the solution of the PML problem. In practice, it is straightforward to implement (i.e., no additional hyperparameter and no need of line search) and it can incorporate many edge-preserving penalty functions.

- In Chapter 4, we develop an accelerated version of the PML algorithm by using the pattern search idea of Hooke and Jeeve [41, pp. 287-291]. Using this approach, we solve a constrained problem at each pattern search step that leads to improved convergence rates. A modification of Hooke and Jeeve's direction vector is also introduced that improves performance. It should be mentioned that Hooke and Jeeve's method has not been used in PET image reconstruction. The proposed algorithm inherits the nice properties of the PML algorithm and converges to the minimizer of the PML objective function. In experiments, the accelerated algorithm needed less than about one third of the CPU-time that was necessary for the PML algorithm to converge.
- In Chapter 5, we propose a regularized image reconstruction algorithm, referred to as the quadratic edge preserving (QEP) algorithm, that aims to preserve edges through the use of certain newly developed de-coupled penalty functions that depend on the current estimate. The QEP algorithm was motivated by the analysis of the PML algorithm. The algorithm by Alenius *et al.* [32] also uses an iteration dependent objective function. However, it should be mentioned that the algorithm uses the OSL algorithm to generate the next iterate. The drawback of Alenius' approach is that the OSL algorithm does not guarantee convergence.
- In Chapter 6, we propose a model for emission data where an unknown matrix, called a scatter matrix, is introduced. The model aims to account for errors due to scatter and non-collinearity. Based on the model, a certain minimization problem is constructed that allows for the scatter matrix and emission mean vector to be jointly estimated. Since the minimization problem is impossible to solve, we propose an algorithm that greatly reduces the number of unknowns in the scatter matrix and alternately estimates the scatter matrix and emission mean vector. It should be mentioned that Mumcuoglu *et al.* [53] used the same

model. However, they assumed that the scatter matrix is known and accounts for errors due to detector penetration as well. Their scatter matrix was obtained through Monte-Carlo simulations.

CHAPTER 3

PENALIZED MAXIMUM LIKELIHOOD ALGORITHM

Although the ML estimates of the emission means are available by using the MLEM algorithm [16], as discussed in Section 1.4, the resulting PET images are extremely noisy due to the fact that the PET image reconstruction problem is ill-posed. This is because of short scan times, errors in the emission data, and the fact that the data obey Poisson statistics. The most popular way to address the ill-posed nature of the PET image reconstruction problem is through the use of penalty functions. Penalty functions used in PET are designed in such a way that estimates for the emission means of neighboring voxels are forced to be similar in value, unless there is an “edge” within neighborhood. By an edge, we mean that there is a group of connected voxels that have significantly greater activity than the other voxels in neighborhood. For example, suppose there is only one voxel with significantly greater activity than the other voxels in its neighborhood. Then, we would say that there is no edge within the neighborhood. Simply stated, penalty functions provide a means for reconstructing PET images that have considerably less noise than MLEM images, yet retain edges (e.g., tumors) which may convey important information.

In Section 3.1, we first derive an algorithm, called the penalized maximum likelihood (PML) algorithm, that incorporates a wide class of edge-preserving penalty functions. Then, we prove that the PML algorithm converges in Section 3.2. Finally, we summarize the properties of the PML algorithm in Section 3.3. It should be mentioned that we presented the PML algorithm in [18] without a proof of convergence. Our proof of convergence can now be found in a recent manuscript [19].

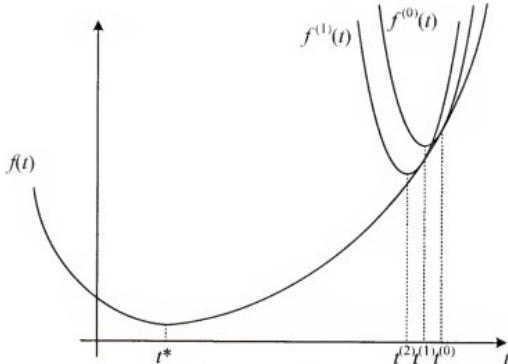


Figure 3.1: One-dimensional illustration of the optimization transfer method. At each iteration, a surrogate function is obtained and a minimizer of the surrogate function is defined as the next iterate. Ideally, it is “easy” to get the minimizer of the surrogate function.

3.1 Penalized Maximum Likelihood (PML) Algorithm

The problem of interest is to determine the nonnegative minimizer of the PML objective function $\Phi(\mathbf{x}) = -\Psi(\mathbf{x}) + \beta\Lambda(\mathbf{x})$ (Ψ is defined in (1.6)), where Λ is a penalty function that forces emission mean estimates of neighboring voxels to be similar in value. In other words, we want to solve the following optimization problem:

$$(P) \quad \mathbf{x}_{\text{PML}} = \arg \min_{\mathbf{x} \geq 0} \Phi(\mathbf{x}) .$$

The penalty functions we consider are of the form

$$\Lambda(\mathbf{x}) = \sum_{j=1}^J \sum_{k \in N_j} \omega_{jk} g(x_j, x_k) , \quad (3.1)$$

where N_j is a set of voxels in a neighborhood of the j^{th} voxel, the constants $\{\omega_{jk}\}$ are positive weights for which $\omega_{jk} = w_{kj}$ for all j and k , and $g(s, t) \stackrel{\Delta}{=} \lambda(s - t)$ whereby the function λ satisfies the following assumptions:

- (AS3) $\lambda(t)$ is symmetric
- (AS4) $\lambda(t)$ is everywhere differentiable

- **(AS5)** $\dot{\lambda}(t) \triangleq \frac{d}{dt}\lambda(t)$ is increasing for all t (assumption implies that λ is strictly convex)
- **(AS6)** $\gamma(t) \triangleq \frac{\dot{\lambda}(t)}{t}$ is nonincreasing for $t \geq 0$
- **(AS7)** $\gamma(0) = \lim_{t \rightarrow 0} \gamma(t)$ is finite and nonzero
- **(AS8)** $\lambda(t)$ is bounded below (assumption implies that $\Lambda(\mathbf{x})$ is bounded below).

Examples of functions that satisfy **(AS3)-(AS8)** are the quadratic function $\lambda(t) = t^2$ and Green's log-cosh function $\lambda(t) = \log(\cosh(t))$ [23]. Regarding the neighborhood N_j , the j^{th} voxel is excluded from the set N_j and, if the k^{th} voxel is in N_j , then the j^{th} voxel is in N_k . A common choice for N_j is the eight nearest neighbors of the j^{th} voxel.

Since it is not possible to get a closed-form solution to the minimization problem **(P)**, iterative optimization methods are necessary. The PML algorithm we propose is based on the optimization transfer method [10, 25, 30, 34, 54] where, at each iteration, a function that satisfies certain conditions is obtained and the next iterate is defined to be a minimizer of the function. The function found at each iteration is referred to as a surrogate function for the function to be minimized. This idea is illustrated with the one-dimensional example in Figure 3.1. In the figure, the problem is to find the minimizer of the function f , which is t^* . It is assumed that a closed-form solution is not available to the minimization problem. Given an initial guess $t^{(0)}$, a surrogate function $f^{(0)}$ that depends on $t^{(0)}$ is determined. Then, the next iterate $t^{(1)}$ is generated by finding the minimizer of $f^{(0)}$. To get the following iterate $t^{(2)}$, a surrogate function $f^{(1)}$ that depends on $t^{(1)}$ is obtained and then minimized. These steps are repeated until some convergence criterion is met.

For a vector argument \mathbf{t} , a surrogate function $f^{(n)}$ satisfies the following conditions:

- **(C1)** $f^{(n)}(\mathbf{t}) \geq f(\mathbf{t})$ for all $\mathbf{t} \in \{\text{domain of } f\}$
- **(C2)** $f^{(n)}(\mathbf{t}^{(n)}) = f(\mathbf{t}^{(n)})$

- **(C3)** $\nabla f^{(n)}(\mathbf{t}^{(n)}) = \nabla f(\mathbf{t}^{(n)})$,

where $\mathbf{t}^{(n)}$ is the n^{th} iterate, ∇ denotes the gradient of a function, and the superscript (n) indicates that the functions $\{f^{(n)}\}$ and the iterates $\{\mathbf{t}^{(n)}\}$ depend on the iteration number. The next iterate $\mathbf{t}^{(n+1)}$ is defined to be a minimizer of $f^{(n)}$:

$$\mathbf{t}^{(n+1)} \triangleq \arg \min_{\mathbf{t}} f^{(n)}(\mathbf{t}) \quad \text{subject to } \mathbf{t} \in \{\text{domain of } f\}. \quad (3.2)$$

Defining the iterates in this way insures that the objective function f decreases monotonically with increasing iterations. To prove this fact, we first note that $f(\mathbf{t}^{(n+1)}) \leq f^{(n)}(\mathbf{t}^{(n+1)})$ by **(C1)**. Since $f^{(n)}(\mathbf{t}^{(n+1)}) \leq f^{(n)}(\mathbf{t}^{(n)})$ by (3.2), it follows by **(C2)** that for all n ,

$$f(\mathbf{t}^{(n+1)}) \leq f^{(n)}(\mathbf{t}^{(n+1)}) \leq f^{(n)}(\mathbf{t}^{(n)}) = f(\mathbf{t}^{(n)}). \quad (3.3)$$

It should be mentioned that **(C3)** is not necessary for the monotonicity in (3.3). However, **(C3)** is often needed in order to prove an algorithm that utilizes the optimization transfer method converges (see [18, 25, 30]). Although the optimization transfer method is straightforward in principle, the difficulty in practice is that it may be difficult to find surrogate functions that satisfy the conditions **(C1)**, **(C2)**, and **(C3)**.

De Pierro developed surrogate functions for the negative log likelihood function $-\Psi$ by using the convexity of the negative log function [25, 30]. His idea is based on the following property of convex functions [55, pp. 860-861]: For a convex function f ,

$$f\left(\sum_j a_j t_j\right) \leq \sum_j a_j f(t_j) \quad (3.4)$$

where $\sum_j a_j t_j \in \{\text{domain of } f\}$, $t_j \in \{\text{domain of } f\}$ for all j , $a_j \geq 0$ for all j , $\sum_j a_j = 1$, and $\{\text{domain of } f\}$ is a convex set. Specifically, De Pierro utilized the

following inequality in ML estimation where $f(t) = -\log(t)$:

$$f([\mathcal{P}\mathbf{x}]_i) = f\left(\sum_j \frac{\mathcal{P}_{ij}x_j^{(n)}}{[\mathcal{P}\mathbf{x}^{(n)}]_i} \frac{[\mathcal{P}\mathbf{x}^{(n)}]_i}{x_j^{(n)}} x_j\right) \quad (3.5)$$

$$\leq \sum_j \frac{\mathcal{P}_{ij}x_j^{(n)}}{[\mathcal{P}\mathbf{x}^{(n)}]_i} f\left(\frac{[\mathcal{P}\mathbf{x}^{(n)}]_i}{x_j^{(n)}} x_j\right), \quad (3.6)$$

where $\mathbf{x}^{(n)}$ is the n^{th} iterate of \mathbf{x} , $\mathcal{P}_{ij} \geq 0$, $x_j^{(n)} \geq 0$, and $[\mathcal{P}\mathbf{x}^{(n)}]_i \neq 0$ for all i, j , and n . Let

$$f_i(\mathbf{x}) \triangleq f([\mathcal{P}\mathbf{x}]_i) \quad (3.7)$$

$$f_i^{(n)}(\mathbf{x}) \triangleq \sum_j \frac{\mathcal{P}_{ij}x_j^{(n)}}{[\mathcal{P}\mathbf{x}^{(n)}]_i} f\left(\frac{[\mathcal{P}\mathbf{x}^{(n)}]_i}{x_j^{(n)}} x_j\right). \quad (3.8)$$

Then, it is straightforward to see that, for all i , (1) $f_i^{(n)}(\mathbf{x}) \geq f_i(\mathbf{x})$ for all $\mathbf{x} > \mathbf{0}$, (2) $f_i^{(n)}(\mathbf{x}^{(n)}) = f_i(\mathbf{x}^{(n)})$, and (3) $\nabla f_i^{(n)}(\mathbf{x}^{(n)}) = \nabla f_i(\mathbf{x}^{(n)})$. Thus, for all i , $f_i^{(n)}$ is a surrogate function for f_i .

Although De Pierro developed the surrogate functions for the log likelihood function under the assumption of no accidental coincidences (i.e., $\boldsymbol{\rho} = \mathbf{0}$), the surrogate functions can be easily modified to account for accidental coincidences. Observe that $[\mathcal{P}\mathbf{x} + \boldsymbol{\rho}]_i$ can be written as a convex combination

$$[\mathcal{P}\mathbf{x} + \boldsymbol{\rho}]_i = \sum_{j=1}^J \frac{\mathcal{P}_{ij}x_j^{(n)}}{[\mathcal{P}\mathbf{x}^{(n)} + \boldsymbol{\rho}]_i} \frac{[\mathcal{P}\mathbf{x}^{(n)} + \boldsymbol{\rho}]_i}{x_j^{(n)}} x_j + \frac{\rho_i}{[\mathcal{P}\mathbf{x}^{(n)} + \boldsymbol{\rho}]_i} [\mathcal{P}\mathbf{x}^{(n)} + \boldsymbol{\rho}]_i. \quad (3.9)$$

Using the convexity of the negative log function and the fact that

$$\sum_{j=1}^J \frac{\mathcal{P}_{ij}x_j^{(n)}}{[\mathcal{P}\mathbf{x}^{(n)} + \boldsymbol{\rho}]_i} + \frac{\rho_i}{[\mathcal{P}\mathbf{x}^{(n)} + \boldsymbol{\rho}]_i} = 1, \quad (3.10)$$

we have the following inequality for $f(t) = -\log(t)$:

$$\begin{aligned} f([\mathcal{P}\mathbf{x} + \boldsymbol{\rho}]_i) &\leq \sum_{j=1}^J \frac{\mathcal{P}_{ij}x_j^{(n)}}{[\mathcal{P}\mathbf{x}^{(n)} + \boldsymbol{\rho}]_i} f\left(\frac{[\mathcal{P}\mathbf{x}^{(n)} + \boldsymbol{\rho}]_i}{x_j^{(n)}} x_j\right) \\ &\quad + \frac{\rho_i}{[\mathcal{P}\mathbf{x}^{(n)} + \boldsymbol{\rho}]_i} f\left([\mathcal{P}\mathbf{x}^{(n)} + \boldsymbol{\rho}]_i\right). \end{aligned} \quad (3.11)$$

Given the inequality in (3.11), the surrogate function at the n^{th} iteration for the negative log likelihood function $-\Psi$ can be expressed as

$$-\Psi^{(n)}(\mathbf{x}) \triangleq \sum_{i=1}^I \left\{ [\mathcal{P}\mathbf{x}]_i - \sum_{j=1}^J d_i \frac{\mathcal{P}_{ij}x_j^{(n)}}{[\mathcal{P}\mathbf{x}^{(n)} + \boldsymbol{\rho}]_i} \log(x_j) \right\} + C_1^{(n)}, \quad (3.12)$$

where

$$C_1^{(n)} = \sum_{i=1}^I \left\{ \rho_i + \log(d_i!) - d_i \log([\mathcal{P}\mathbf{x}^{(n)} + \boldsymbol{\rho}]_i) + \sum_{j=1}^J d_i \frac{\mathcal{P}_{ij}x_j^{(n)}}{[\mathcal{P}\mathbf{x}^{(n)}]_i + \rho_i} \log(x_j^{(n)}) \right\}. \quad (3.13)$$

It is straightforward to show that the surrogate function $\Psi^{(n)}$ satisfies **(C1)**, **(C2)**, and **(C3)**: (1) $\Psi^{(n)}(\mathbf{x}) \leq \Psi(\mathbf{x})$ for all $\mathbf{x} > \mathbf{0}$, (2) $\Psi^{(n)}(\mathbf{x}^{(n)}) = \Psi(\mathbf{x}^{(n)})$, and (3) $\nabla\Psi^{(n)}(\mathbf{x}^{(n)}) = \nabla\Psi(\mathbf{x}^{(n)})$.

Since surrogate functions for the negative log likelihood function $-\Psi$ are available, we only need to find surrogate functions for the penalty function $\Lambda(\mathbf{x}) = \sum_j \sum_k \omega_{jk} \lambda(x_j - x_k)$. Under assumptions **(AS3)-(AS8)**, Huber developed a surrogate function for λ in [44] (see also [8]). Given an arbitrary point $t^{(n)}$, Huber's surrogate function for λ , which is defined by

$$\bar{\lambda}^{(n)}(t) \triangleq \lambda(t^{(n)}) + \dot{\lambda}(t^{(n)})(t - t^{(n)}) + \frac{1}{2}\gamma(t^{(n)})(t - t^{(n)})^2, \quad (3.14)$$

has the property that $\bar{\lambda}^{(n)}(t) \geq \lambda(t)$ for all t (see Appendix A), $\bar{\lambda}^{(n)}(t^{(n)}) = \lambda(t^{(n)})$, and $\dot{\bar{\lambda}}^{(n)}(t^{(n)}) = \dot{\lambda}(t^{(n)})$, where the dot over a function represents its first derivative.

For $t^{(n)} \triangleq x_j^{(n)} - x_k^{(n)}$, it follows that a surrogate function for Λ is

$$\bar{\Lambda}^{(n)}(\mathbf{x}) = \sum_{j=1}^J \sum_{k \in N_j} \omega_{jk} \bar{g}^{(n)}(x_j, x_k), \quad (3.15)$$

where $\bar{g}^{(n)}(s, t) \triangleq \bar{\lambda}^{(n)}(s - t)$.

Using $\bar{\lambda}^{(n)}$ as a starting point, we will now construct a surrogate function for Λ that has a more convenient form. By the convexity of the square function, we have

the following inequality

$$\begin{aligned} [x_j - x_k - (x_j^{(n)} - x_k^{(n)})]^2 &= \left[\frac{1}{2}(2x_j - 2x_j^{(n)}) + \frac{1}{2}(2x_k^{(n)} - 2x_k) \right]^2 \\ &\leq \frac{1}{2}(2x_j - 2x_j^{(n)})^2 + \frac{1}{2}(2x_k^{(n)} - 2x_k)^2. \end{aligned} \quad (3.16)$$

It should be mentioned that De Pierro [30] and Hsiao *et al.* [34] utilized this property of the quadratic function in the PET, and Erdo\u{g}an and Fessler [10] applied it to a non-quadratic convex function for transmission tomography. Motivated by (3.16), we define

$$\begin{aligned} g^{(n)}(x_j, x_k) &\triangleq \lambda(x_j^{(n)} - x_k^{(n)}) + \dot{\lambda}(x_j^{(n)} - x_k^{(n)}) \left[(x_j - x_j^{(n)}) - (x_k - x_k^{(n)}) \right] \\ &\quad + \frac{1}{4}\gamma(x_j^{(n)} - x_k^{(n)}) \left[(2x_j - 2x_j^{(n)})^2 + (2x_k - 2x_k^{(n)})^2 \right]. \end{aligned} \quad (3.17)$$

By construction, the following statements are clear: (1) $g^{(n)}(x_j, x_k) \geq g(x_j, x_k)$ for all x_j and x_k from (3.16), (2) $g^{(n)}(x_j^{(n)}, x_k^{(n)}) = g(x_j^{(n)}, x_k^{(n)})$, (3) $\frac{\partial}{\partial x_j} g^{(n)}(x_j^{(n)}, x_k^{(n)}) = \frac{\partial}{\partial x_j} g(x_j^{(n)}, x_k^{(n)})$, and (4) $\frac{\partial}{\partial x_k} g^{(n)}(x_j^{(n)}, x_k^{(n)}) = \frac{\partial}{\partial x_k} g(x_j^{(n)}, x_k^{(n)})$. The difference between $g^{(n)}$ and $\bar{g}^{(n)}$ is that $g^{(n)}$ is de-coupled in the sense that it does not have terms of the form $x_j x_k$. This difference is important because, as we show later, using $g^{(n)}$ enables us to construct surrogate functions for Φ that have closed form expressions for their minimizers. Using $g^{(n)}$, an alternative surrogate function for Λ is

$$\Lambda^{(n)}(\mathbf{x}) = \sum_{j=1}^J \sum_{k \in N_j} \omega_{jk} g^{(n)}(x_j, x_k). \quad (3.18)$$

It is clear that, by construction, $\Lambda^{(n)}$ satisfies the following properties: (1) $\Lambda^{(n)}(\mathbf{x}) \geq \Lambda(\mathbf{x})$ for all \mathbf{x} , (2) $\Lambda^{(n)}(\mathbf{x}^{(n)}) = \Lambda(\mathbf{x}^{(n)})$, and (3) $\nabla \Lambda^{(n)}(\mathbf{x}^{(n)}) = \nabla \Lambda(\mathbf{x}^{(n)})$.

Now, using $\Psi^{(n)}$ and $\Lambda^{(n)}$, the desired surrogate function at the n^{th} iteration for Φ is

$$\Phi^{(n)}(\mathbf{x}) = -\Psi^{(n)}(\mathbf{x}) + \beta \Lambda^{(n)}(\mathbf{x}). \quad (3.19)$$

From the properties of $\Psi^{(n)}$ and $\Lambda^{(n)}$, it follows that the surrogate function $\Phi^{(n)}$ possesses the requisite properties:

- (P1) $\Phi^{(n)}(\mathbf{x}) \geq \Phi(\mathbf{x})$ for all $\mathbf{x} > \mathbf{0}$
- (P2) $\Phi^{(n)}(\mathbf{x}^{(n)}) = \Phi(\mathbf{x}^{(n)})$
- (P3) $\nabla \Phi^{(n)}(\mathbf{x}^{(n)}) = \nabla \Phi(\mathbf{x}^{(n)})$.

Given $\mathbf{x}^{(n)}$, the next iterate $\mathbf{x}^{(n+1)}$ is found by minimizing $\Phi^{(n)}$:

$$\mathbf{x}^{(n+1)} \triangleq \arg \min_{\mathbf{x} \geq \mathbf{0}} \Phi^{(n)}(\mathbf{x}) . \quad (3.20)$$

Defining the iterates in this way insures that the objective function Φ decreases with increasing iterations as shown in (3.3):

$$\Phi(\mathbf{x}^{(n)}) \geq \Phi(\mathbf{x}^{(n+1)}) \quad \text{for all } n . \quad (3.21)$$

All that remains now is to solve the optimization problem in (3.20). To this end, we write $\Lambda^{(n)}(\mathbf{x})$ as

$$\Lambda^{(n)}(\mathbf{x}) = 2 \sum_{j=1}^J \sum_{k \in N_j} h_{jk}^{(n)}(x_j) + C_2^{(n)} , \quad (3.22)$$

where

$$h_{jk}^{(n)}(t) \triangleq \omega_{jk} \gamma(x_j^{(n)} - x_k^{(n)})(t - m_{jk}^{(n)})^2 \quad (3.23)$$

$$m_{jk}^{(n)} \triangleq \frac{x_j^{(n)} + x_k^{(n)}}{2} \quad (3.24)$$

$$C_2^{(n)} \triangleq \sum_{j=1}^J \sum_{k \in N_j} \omega_{jk} \left\{ \lambda(x_j^{(n)} - x_k^{(n)}) - \frac{1}{2} \dot{\lambda}(x_j^{(n)} - x_k^{(n)})(x_j^{(n)} - x_k^{(n)}) \right\} \quad (3.25)$$

(see proof in Appendix B). Since $\Psi^{(n)}$ and $\Lambda^{(n)}$ are de-coupled, $\Phi^{(n)}$ can be written as

$$\Phi^{(n)}(\mathbf{x}) = -\Psi^{(n)}(\mathbf{x}) + \beta\Lambda^{(n)}(\mathbf{x}) \quad (3.26)$$

$$\begin{aligned} &= \sum_{i=1}^I \left\{ [\mathcal{P}\mathbf{x}]_i - \sum_{j=1}^J d_i \frac{\mathcal{P}_{ij}x_j^{(n)}}{[\mathcal{P}\mathbf{x}^{(n)} + \boldsymbol{\rho}]_i} \log(x_j) \right\} \\ &\quad + 2\beta \sum_{j=1}^J \sum_{k \in N_j} h_{jk}^{(n)}(x_j) + C_1^{(n)} + \beta C_2^{(n)} \end{aligned} \quad (3.27)$$

$$\begin{aligned} &= \sum_{j=1}^J \left\{ x_j \sum_{i=1}^I \mathcal{P}_{ij} - \log(x_j) \sum_{i=1}^I d_i \frac{\mathcal{P}_{ij}x_j^{(n)}}{[\mathcal{P}\mathbf{x}^{(n)} + \boldsymbol{\rho}]_i} \right\} + C_1^{(n)} + \beta C_2^{(n)} \\ &\quad + \sum_{j=1}^J \left\{ 2\beta \sum_{k \in N_j} \omega_{jk} \gamma(x_j^{(n)} - x_k^{(n)}) (x_j^2 - 2m_{jk}^{(n)}x_j + m_{jk}^{(n)2}) \right\} \end{aligned} \quad (3.28)$$

$$= \sum_{j=1}^J \{E_j^{(n)} \log(x_j) + F_j^{(n)}x_j^2 + G_j^{(n)}x_j\} + C_3^{(n)}, \quad (3.29)$$

where

$$E_j^{(n)} \triangleq -\sum_{i=1}^I d_i \frac{\mathcal{P}_{ij}x_j^{(n)}}{[\mathcal{P}\mathbf{x}^{(n)} + \boldsymbol{\rho}]_i} \quad (3.30)$$

$$F_j^{(n)} \triangleq 2\beta \sum_{k \in N_j} \omega_{jk} \gamma(x_j^{(n)} - x_k^{(n)}) \quad (3.31)$$

$$G_j^{(n)} \triangleq \sum_{i=1}^I \mathcal{P}_{ij} - 4\beta \sum_{k \in N_j} \omega_{jk} \gamma(x_j^{(n)} - x_k^{(n)}) m_{jk}^{(n)} \quad (3.32)$$

$$C_3^{(n)} = C_1^{(n)} + \beta C_2^{(n)} + 2\beta \sum_{j=1}^J \sum_{k \in N_j} \omega_{jk} \gamma(x_j^{(n)} - x_k^{(n)}) m_{jk}^{(n)2}. \quad (3.33)$$

Since $\Phi^{(n)}$ is de-coupled, as seen from (3.29), it follows that the solution to (3.20) is given by

$$x_j^{(n+1)} = \arg \min_{x_j \geq 0} \phi_j^{(n)}(x_j), \quad j = 1, 2, \dots, J. \quad (3.34)$$

where

$$\phi_j^{(n)}(t) \triangleq E_j^{(n)} \log(t) + F_j^{(n)}t^2 + G_j^{(n)}t. \quad (3.35)$$

Fortunately, the function $\phi_j^{(n)}$ is strictly convex for all j and n under the assumption that $x_j^{(n)} > 0$ for all j and n . We will prove this statement by showing that the second derivative of $\phi_j^{(n)}$ is positive, when $x_j^{(n)} > 0$ for all j and n . First, note that $E_j^{(n)}$ is negative and $F_j^{(n)}$ is positive for all j and n . The fact that $E_j^{(n)} > 0$ is due to the fact that $d'P_j \neq 0$ (see **(AS2)** in Section 1.4) and the assumption that $x_j^{(n)} > 0$. The fact that $F_j^{(n)} > 0$ follows from the positivity of the function γ , weights $\{\omega_{jk}\}$, and penalty parameter β . To see why $\gamma(t) > 0$ for $-\infty < t < \infty$, recall that $\lambda(t)$ is a symmetric, strictly convex function by **(AS3)** and **(AS5)**. It follows that $\dot{\lambda}(t) > 0$ over $(0, \infty)$ and $\dot{\lambda}(t) < 0$ over $(-\infty, 0)$. Using the fact that $\gamma(0)$ is finite and nonzero (see **(AS7)**), we have that $\gamma(t) > 0$ for $-\infty < t < \infty$. Now, consider the second derivative of $\phi_j^{(n)}$. Easy calculations show that $\ddot{\phi}_j^{(n)}(t) = (-E_j^{(n)}/t^2 + 2F_j^{(n)})$, where the double dot over a function represents the second derivative of a function. Since $F_j^{(n)} > 0$ and $E_j^{(n)} < 0$, it follows that the second derivative of $\phi_j^{(n)}(t)$ is positive for all j and n . Thus, $\phi_j^{(n)}(t)$ is strictly convex for all j , n , and $t > 0$, and, from (3.29), $\Phi^{(n)}$ is strictly convex over the set $\{\mathbf{x} : \mathbf{x} > \mathbf{0}\}$.

Since $\Phi^{(n)}$ is de-coupled, $\phi_j^{(n)}$ is strictly convex, and $\phi_j^{(n)}(t) \rightarrow \infty$ as $t \rightarrow 0^+$, it is true that

$$\mathbf{x}^{(n+1)} > \mathbf{0} \quad \text{and} \quad \dot{\phi}_j^{(n)}(x_j^{(n+1)}) = 0. \quad (3.36)$$

Note that (3.36) satisfies our assumption that $x_j^{(n)} > 0$ for all j and n . To solve (3.34), we compute the first derivative of $\phi_j^{(n)}$ and set it to zero. Since $E_j^{(n)} < 0$ and $F_j^{(n)} > 0$, the root of the quadratic equation that preserves the nonnegativity constraint is

$$x_j^{(n+1)} = \frac{-G_j^{(n)} + \sqrt{G_j^{(n)2} - 8E_j^{(n)}F_j^{(n)}}}{4F_j^{(n)}}, \quad j = 1, 2, \dots, J. \quad (3.37)$$

Observe, as $\beta \rightarrow 0$, (3.37) approaches $-E_j^{(n)} / \sum_i P_{ij}$ by L'Hospital's rule. Thus, the iteration in (3.37) is equivalent to the MLEM algorithm when $\beta = 0$.

In summary, given a strictly positive initial estimate $\mathbf{x}^{(0)} > \mathbf{0}$, the steps of the PML algorithm are: for $n = 0, 1, 2, \dots$

- **Step 1** Let $\mathbf{x}^{(0)} > \mathbf{0}$ be the initial estimate
- **Step 2** Construct the surrogate function $\Phi^{(n)}$ from the current iterate $\mathbf{x}^{(n)}$ using (3.29), (3.30), (3.31), and (3.32)
- **Step 3** Get \mathbf{x}^{n+1} using (3.37)
- **Step 4** Iterate between **Steps 2** and **3** until some chosen stopping criterion is met.

3.2 Convergence Proof

Using **(P1)-(P3)** and **(AS1)-(AS8)**, we now prove that the PML algorithm converges. The following convergence proof is based on the convergence proof by Lange and Carson [56] (see also [30]) of the MLEM algorithm by Shepp and Vardi [16].

By (3.21), the PML algorithm has the property that it decreases the objective function Φ with increasing iterations:

- **(P4)** $\Phi(\mathbf{x}^{(n+1)}) \leq \Phi(\mathbf{x}^{(n)})$ for all $n \geq 0$.

Another property of the algorithm is that

- **(P5)** the sequence $\{\Phi(\mathbf{x}^{(n)})\}$ is convergent.

This property follows from **(P4)** and the fact that Φ is bounded below by **(AS8)** (see [57, Theorem 1.4, p. 6]).

Proposition 1 *The sequence $\{\mathbf{x}^{(n)}\}$ is bounded.*

Proof : From **(P4)**, it follows that $\Phi(\mathbf{x}^{(n)}) \leq \Phi(\mathbf{x}^{(0)})$ for all $n \geq 0$. Consider the set $\mathbb{B} \triangleq \{\mathbf{x} \geq \mathbf{0} : \Phi(\mathbf{x}) \leq \Phi(\mathbf{x}^{(0)})\}$. Then, clearly $\{\mathbf{x}^{(n)}\} \subset \mathbb{B}$. So, to prove $\{\mathbf{x}^{(n)}\}$ is bounded, we will prove that the set \mathbb{B} is bounded. It is straightforward to see that \mathbb{B} is bounded below by $\mathbf{0}$. Now, suppose that \mathbb{B} is not bounded above. Then, there exists a point $\mathbf{z} \in \mathbb{B}$ such that $\|\mathbf{z}\| \not\leq \infty$. This result means that for some j there exists z_j such that $z_j \not\leq \infty$. Since **(AS2)** implies that \mathcal{P} has no column vector of zeros, it follows that $[\mathcal{P}\mathbf{z}]_i \not\leq \infty$ for some i and $\Phi(\mathbf{z}) \not\leq \infty$. This implies that \mathbf{z} is not

an element of the set \mathbb{B} because all the elements of \mathbb{B} have an objective value that is less than or equal to $\Phi(\mathbf{x}^{(0)}) < \infty$, which is a contradiction. Therefore, the set \mathbb{B} is bounded above. ■

Proposition 2 *There exists some constant $c_1 > 0$ such that $\Phi(\mathbf{x}^{(n)}) - \Phi(\mathbf{x}^{(n+1)}) \geq c_1 \|\mathbf{x}^{(n)} - \mathbf{x}^{(n+1)}\|^2$ for all $n \geq 0$.*

Proof : By (P1), (P2), and (3.29), we have the following inequality

$$\Phi(\mathbf{x}^{(n)}) - \Phi(\mathbf{x}^{(n+1)}) \geq \Phi^{(n)}(\mathbf{x}^{(n)}) - \Phi^{(n)}(\mathbf{x}^{(n+1)}) = \sum_{j=1}^J \left\{ \phi_j^{(n)}(x_j^{(n)}) - \phi_j^{(n)}(x_j^{(n+1)}) \right\}. \quad (3.38)$$

Suppose, for each n , the function $\phi_j^{(n)}(t)$ is expanded into a second-order Taylor series [55, pp. 868-869] about the point $x_j^{(n+1)}$ and evaluated at $t = x_j^{(n)}$. Then, the right hand side of (3.38) can be written as

$$\Phi^{(n)}(\mathbf{x}^{(n)}) - \Phi^{(n)}(\mathbf{x}^{(n+1)}) = \sum_{j=1}^J \left\{ (x_j^{(n)} - x_j^{(n+1)}) \dot{\phi}_j^{(n)}(x_j^{(n+1)}) + \frac{1}{2} (x_j^{(n)} - x_j^{(n+1)})^2 \ddot{\phi}_j^{(n)}(\bar{x}_j^{(n+1)}) \right\}, \quad (3.39)$$

where the double dot over a function represents its second derivative and $\bar{x}_j^{(n+1)}$ is a point between $x_j^{(n)}$ and $x_j^{(n+1)}$. Since $\dot{\phi}_j^{(n)}(x_j^{(n+1)}) = 0$ by (3.36), it follows

$$\Phi^{(n)}(\mathbf{x}^{(n)}) - \Phi^{(n)}(\mathbf{x}^{(n+1)}) = \sum_{j=1}^J \frac{1}{2} (x_j^{(n)} - x_j^{(n+1)})^2 \ddot{\phi}_j^{(n)}(\bar{x}_j^{(n+1)}). \quad (3.40)$$

Now, recall that $\ddot{\phi}_j^{(n)}(t) = (-E_j^{(n)}/t^2 + 2F_j^{(n)})$ with $F_j^{(n)} = 2\beta \sum_{k \in N_j} \omega_{jk} \gamma(x_j^{(n)} - x_k^{(n)})$ and $E_j^{(n)} < 0$. Since $\{\mathbf{x}^{(n)}\}$ is bounded and $\gamma(t) > 0$ is a continuous function for $-\infty < t < \infty$, there exists a number $\gamma_0 > 0$ such that $\gamma(x_j^{(n)} - x_k^{(n)}) \geq \gamma_0$ for all j, k , and n . Hence, $F_j^{(n)} \geq c_1$ for all j and n , where $c_1 = 2\beta\gamma_0 \min_j \sum_{k \in N_j} \omega_{jk} > 0$. Therefore, $\ddot{\phi}_j^{(n)}(\bar{x}_j) \geq 2c_1$ for all j and n , and we obtain the desired result

$$\Phi(\mathbf{x}^{(n)}) - \Phi(\mathbf{x}^{(n+1)}) \geq c_1 \|\mathbf{x}^{(n)} - \mathbf{x}^{(n+1)}\|^2. \quad (3.41)$$

■

From (P5) and *Proposition 2*, it follows that

- (P6) the sequence $\{\mathbf{x}^{(n)} - \mathbf{x}^{(n+1)}\}$ converges to 0.

The following proposition will be used later to prove not only that a limit point of the sequence $\{\mathbf{x}^{(n)}\}$ satisfies one of the Kuhn-Tucker conditions [55, p. 777] but also that the sequence $\{\mathbf{x}^{(n)}\}$ has a finite number of limit points.

Proposition 3 *Let \mathbf{x}^* be a limit point of the sequence $\{\mathbf{x}^{(n)}\}$. Then, for all j such that $x_j^* \neq 0$,*

$$\frac{\partial}{\partial x_j} \Phi(\mathbf{x}) \Big|_{\mathbf{x}=\mathbf{x}^*} = 0. \quad (3.42)$$

Proof : By *Proposition 1*, there is a subsequence $\{\mathbf{x}^{(n_l)}\}$ that converges to \mathbf{x}^* (see the Bolzano-Weierstrass theorem in [58, p. 108]). By (P6), the subsequence $\{\mathbf{x}^{(n_l+1)}\}$ also converges to \mathbf{x}^* . Recall from (3.29) that $\dot{\phi}_j^{(n_l)}(t) = E_j^{(n_l)}/t + 2F_j^{(n_l)}t + G_j^{(n_l)}$. If $x_j^* \neq 0$, it follows that $E_j^{(n_l)}/x_j^{(n_l)}$ and $E_j^{(n_l)}/x_j^{(n_l+1)}$ converge to the same limit, and hence

$$\lim_{l \rightarrow \infty} \dot{\phi}_j^{(n_l)}(x_j^{(n_l)}) = \lim_{l \rightarrow \infty} \dot{\phi}_j^{(n_l)}(x_j^{(n_l+1)}). \quad (3.43)$$

Since $\dot{\phi}_j^{(n_l)}(x_j^{(n_l+1)}) = 0$ by (3.36), it follows from (P3) that

$$\frac{\partial}{\partial x_j} \Phi(\mathbf{x}) \Big|_{\mathbf{x}=\mathbf{x}^*} = 0 \quad \text{for all } j \text{ such that } x_j^* \neq 0. \quad (3.44)$$

■

Using *Proposition 3*, we can prove the following proposition, which will be used to prove that the sequence $\{\mathbf{x}^{(n)}\}$ converges.

Proposition 4 *The sequence $\{\mathbf{x}^{(n)}\}$ has a finite number of limit points.*

Proof : Consider the following sets

$$\mathbb{Y} \triangleq \{1, 2, \dots, J\} \quad (3.45)$$

$$\mathbb{Z}^* \triangleq \{j \in \mathbb{Y} : x_j^* = 0\} \quad (3.46)$$

$$\mathbb{Z}^{**} \triangleq \{j \in \mathbb{Y} : x_j^{**} = 0\}, \quad (3.47)$$

where \mathbf{x}^* and \mathbf{x}^{**} are limit points of $\{\mathbf{x}^{(n)}\}$. Now, let the function $\Phi^*(\mathbf{x})$ be the restriction of $\Phi(\mathbf{x})$ to the reduced parameter set $\mathbb{W}^* \triangleq \{\mathbf{x} \geq \mathbf{0} : x_j = 0 \text{ for } j \in \mathbb{Z}^*\}$. Since $\Phi^*(\mathbf{x})$ is strictly convex over a convex set, the unique minimizer of $\Phi^*(\mathbf{x})$ is its only stationary point [59, Proposition 2.1.2, p. 87]. Thus, by *Proposition 3*, there is only one limit point of $\{\mathbf{x}^{(n)}\}$ in the set \mathbb{W}^* . In other words, if $\mathbb{Z}^* = \mathbb{Z}^{**}$, then $\mathbf{x}^* = \mathbf{x}^{**}$. Therefore, the number of limit points are bounded above by the number of subsets of \mathbb{Y} , which is clearly finite. ■

Theorem 1 *The sequence $\{\mathbf{x}^{(n)}\}$ converges to the unique minimizer of Φ .*

Proof: Let \mathbf{x}^* be a limit point of the sequence $\{\mathbf{x}^{(n)}\}$. Using the theorem in [60, p. 173], which says that the set of limit points of a bounded sequence $\{\mathbf{x}^{(n)}\}$ is connected if $\{\mathbf{x}^{(n)} - \mathbf{x}^{(n+1)}\} \rightarrow 0$, we obtain the fact that the set of limit points of $\{\mathbf{x}^{(n)}\}$ is connected by *Proposition 1* and **(P6)**. Since the number of limit points of $\{\mathbf{x}^{(n)}\}$ is finite by *Proposition 4*, $\{\mathbf{x}^{(n)}\}$ has only one limit point. Thus, $\{\mathbf{x}^{(n)}\} \rightarrow \mathbf{x}^*$. Now, note that the PML objective function Φ is strictly convex on the set $\{\mathbf{x} : \mathbf{x} \geq \mathbf{0}\}$ (see Appendix C), so there is only one minimizer. To prove the sequence $\{\mathbf{x}^{(n)}\}$ converges to the unique minimizer, we need to show that \mathbf{x}^* satisfies the Kuhn-Tucker conditions [55, p. 777]: for all j ,

$$x_j^* \geq 0 \quad (3.48)$$

$$x_j^* \frac{\partial}{\partial x_j} \Phi(\mathbf{x}) \Big|_{\mathbf{x}=\mathbf{x}^*} = 0 \quad (3.49)$$

$$\frac{\partial}{\partial x_j} \Phi(\mathbf{x}) \Big|_{\mathbf{x}=\mathbf{x}^*} \geq 0. \quad (3.50)$$

Since $\mathbf{x}^{(n)} > \mathbf{0}$ for all n , it must be true that the limit point \mathbf{x}^* is nonnegative (i.e., (3.48) is satisfied). By *Proposition 3*,

$$\frac{\partial}{\partial x_j} \Phi(\mathbf{x}) \Big|_{\mathbf{x}=\mathbf{x}^*} = 0 \quad \text{for all } j \text{ such that } x_j^* \neq 0. \quad (3.51)$$

So, (3.49) is satisfied. Now, we consider the case $x_j^* = 0$. For j such that $x_j^* = 0$, suppose

$$\frac{\partial}{\partial x_j} \Phi(\mathbf{x}) \Big|_{\mathbf{x}=\mathbf{x}^*} < 0. \quad (3.52)$$

Then, it follows that $\lim_{n \rightarrow \infty} \dot{\phi}_j^{(n)}(x_j^{(n)}) < 0$ by (P3), and $\dot{\phi}_j^{(n)}(x_j^{(n)}) < 0$ for sufficiently large n . Consider $\dot{\phi}_j^{(n)}(x_j^{(n+1)}) = E_j^{(n)} / x_j^{(n+1)} + 2F_j^{(n)} x_j^{(n+1)} + G_j^{(n)}$. If $x_j^{(n+1)} < x_j^{(n)}$, then

$$\frac{x_j^{(n+1)}}{x_j^{(n)}} \frac{E_j^{(n)}}{x_j^{(n+1)}} + 2F_j^{(n)} x_j^{(n+1)} + G_j^{(n)} > 0 \quad (3.53)$$

because $\dot{\phi}_j^{(n)}(x_j^{(n+1)}) = 0$ by (3.36) and $E_j^{(n)} / x_j^{(n+1)} < 0$. Moreover, the fact $F_j^{(n)} > 0$ implies that

$$\frac{E_j^{(n)}}{x_j^{(n)}} + 2F_j^{(n)} x_j^{(n)} + G_j^{(n)} = \dot{\phi}_j^{(n)}(x_j^{(n)}) > 0, \quad (3.54)$$

which is a contradiction. Thus, $x_j^{(n+1)} \geq x_j^{(n)}$ for all sufficiently large n . However, this contradicts the fact $x_j^{(n)} \rightarrow 0$. So, it is true that

$$\frac{\partial}{\partial x_j} \Phi(\mathbf{x}) \Big|_{\mathbf{x}=\mathbf{x}^*} \geq 0 \quad \text{for all } j \text{ such that } x_j^* = 0. \quad (3.55)$$

This satisfies (3.50). ■

3.3 Properties of the PML Algorithm

We now provide a summary of the desirable properties of the PML algorithm:

- The PML algorithm is straightforward to implement because there are no hyperparameters required for the algorithm itself and it has closed-form expressions for the iterates. Some algorithms require hyperparameters, such as relaxation parameters, in addition to the penalty parameter [33, 35], while others [24, 30] do not have closed-form expressions for the updates.
- The PML algorithm theoretically guarantees nonnegative iterates, whereas some algorithms [33, 35] set any negative element of the iterates to a small positive number.

- The PML algorithm monotonically decreases the PML objective function unlike the algorithms in [23, 35].
 - The PML algorithm can incorporate a large class of edge-preserving penalty functions unlike the algorithm by De Pierro [30].
 - The PML algorithm converges to the minimizer of the PML objective function.
- Convergence proofs for the algorithms in [23, 33] are not available.

CHAPTER 4

ACCELERATED PENALIZED MAXIMUM LIKELIHOOD ALGORITHM

Although the PML algorithm presented in Chapter 3 converges to the nonnegative minimizer of the PML objective function, it has the drawback that it converges slowly. In PET, a popular way to accelerate iterative image reconstruction algorithms is through the use of so called ordered-subsets [45]. In ordered-subsets based reconstruction algorithms, the observed data, \mathbf{d} , is divided into a predefined number of subsets via some chosen rule. Then, the iterative reconstruction algorithm to be accelerated is applied sequentially to each data subset. In [45], Hudson and Larkin developed the first PET image reconstruction algorithm that used the ordered-subsets idea. Since the MLEM algorithm was applied to each data subset, they called their algorithm the ordered-subsets expectation maximization (OS-EM) algorithm. In [61], Browne and De Pierro showed that the OS-EM algorithm did not converge and introduced another ordered-subsets based image reconstruction algorithm that employed a relaxation parameter. It should be pointed out that some convergence results are available for ordered subsets based algorithms that use relaxation parameters [33, 35, 61].

With ordered-subsets based algorithms, there is uncertainty as to how many subsets to be used and how the data should be divided. Moreover, it is not clear how relaxation parameters should be chosen in practice because, generally, they depend on the data.

In this chapter, we introduce an accelerated version of the PML algorithm, referred to as the accelerated PML (APML) algorithm, that uses a pattern search suggested by Hooke and Jeeve [41, pp. 287-291]. A pattern search has also been exploited to accelerate an algorithm in the transmission tomography [62]. In Section

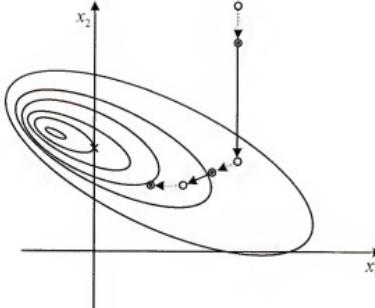


Figure 4.1: Two-dimensional illustration of the sequence $\{\chi^{(n)}\}$. The single circles and double circles denote the accelerated iterates $\{\tilde{x}^{(n)}\}$ and standard iterates $\{x^{(n)}\}$, respectively. Each ellipse represents a set of points that have same cost. The mark x denotes the minimizer of the function subject to the constraints $x_1 \geq 0$ and $x_2 \geq 0$.

4.1, using the mathematical ideas in the convergence proof of the PML algorithm, we show that a sequence that satisfies certain conditions converges to the minimizer of the PML objective function. Then, we use this result to prove that the APML algorithm, which is developed in Section 4.2, converges to nonnegative minimizer of the PML objective function. In Section 4.3, we introduce the direction vector to be used in the pattern search. Finally, we summarize the properties of the APML algorithm in Section 4.4. It should be mentioned that we introduced the APML algorithm in [19].

4.1 Convergence Proof

In this section, we prove that the sequence $\{\chi^{(n)}\} \triangleq \{\tilde{x}^{(0)}, x^{(1)}, \tilde{x}^{(1)}, x^{(2)}, \tilde{x}^{(2)}, \dots\}$ converges to the minimizer of the PML objective function Φ , where $\tilde{x}^{(0)} > \mathbf{0}$ is an initial guess, $x^{(n+1)} \triangleq \arg \min \tilde{\Phi}^{(n)}(\mathbf{x})$ subject to $\mathbf{x} \geq \mathbf{0}$, $\tilde{\Phi}^{(n)}$ is of the same form of the surrogate function for the PML objective function Φ at the iterate $x^{(n)}$, $\Phi^{(n)}$ in (3.29), except that $\tilde{\Phi}^{(n)}$ is defined at the point $\tilde{x}^{(n)}$ instead of $x^{(n)}$, and the point $\tilde{x}^{(n)} > \mathbf{0}$ satisfies the following conditions for all n

- (C4) $\Phi(\tilde{x}^{(n)}) \leq \Phi(x^{(n)})$

- (C5) there exists some constant $c_2 > 0$ such that

$$\Phi(\mathbf{x}^{(n)}) - \Phi(\tilde{\mathbf{x}}^{(n)}) \geq c_2 \|\mathbf{x}^{(n)} - \tilde{\mathbf{x}}^{(n)}\|^2.$$

This convergence result will form the basis for the convergence proof of the APML algorithm. Note, the strict positivity of $\tilde{\mathbf{x}}^{(n)}$ for all n is necessary because the surrogate function for the PML objective function, $\tilde{\Phi}^{(n)}$, is undefined for vectors with zero or negative elements. An example of such a sequence $\{\chi^{(n)}\}$ is illustrated in Figure 4.1. In the figure, the single circles and double circles represent the accelerated iterates $\{\tilde{\mathbf{x}}^{(n)}\}$ and standard iterates $\{\mathbf{x}^{(n)}\}$, respectively.

First, note that the following convergence proof is mainly based on the convergence proof of the PML algorithm in Chapter 3. Since Φ is bounded below and the sequence $\{\chi^{(n)}\}$ monotonically decreases the PML objective function Φ by (P4) and (C4), it follows that (see [57, Theorem 1.4, p. 6])

- (P7) the sequence $\{\Phi(\chi^{(n)})\}$ converges.

By (P4) and (C4), it is straightforward to see that $\{\chi^{(n)}\} \subset \tilde{\mathbb{B}} = \{\mathbf{x} \geq \mathbf{0} : \Phi(\mathbf{x}) \leq \Phi(\tilde{\mathbf{x}}^{(0)})\}$. Since the set $\tilde{\mathbb{B}}$ is bounded (see *Proposition 1*), it follows that

- (P8) the sequence $\{\chi^{(n)}\}$ is bounded.

Now, note that $\mathbf{x}^{(n+1)}$ is the minimizer of the surrogate function $\tilde{\Phi}^{(n)}$, which satisfies (1) $\tilde{\Phi}^{(n)}(\mathbf{x}) \geq \Phi(\mathbf{x})$ for all $\mathbf{x} > \mathbf{0}$, (2) $\tilde{\Phi}^{(n)}(\tilde{\mathbf{x}}^{(n)}) = \Phi(\tilde{\mathbf{x}}^{(n)})$, and (3) $\nabla \tilde{\Phi}^{(n)}(\tilde{\mathbf{x}}^{(n)}) = \nabla \Phi(\tilde{\mathbf{x}}^{(n)})$. Thus, by (P8) (see *Proposition 2*), we have the following property:

- (P9) there exists some constant $c_3 > 0$ such that $\Phi(\tilde{\mathbf{x}}^{(n)}) - \Phi(\mathbf{x}^{(n+1)}) \geq c_3 \|\tilde{\mathbf{x}}^{(n)} - \mathbf{x}^{(n+1)}\|^2$.

By (P7) and (P9), we obtain the property

- (P10) that the sequence $\{\tilde{\mathbf{x}}^{(n)} - \mathbf{x}^{(n+1)}\}$ converges to 0.

Also, by (P7) and (C5), it must be true that the sequence $\{\mathbf{x}^{(n)} - \tilde{\mathbf{x}}^{(n)}\}$ converges to 0. Thus, from the fact that $\|\tilde{\mathbf{x}}^{(n)} - \tilde{\mathbf{x}}^{(n+1)}\|^2 \leq \|\tilde{\mathbf{x}}^{(n)} - \mathbf{x}^{(n+1)}\|^2 + \|\mathbf{x}^{(n+1)} - \tilde{\mathbf{x}}^{(n+1)}\|^2$, the property follows:

- (P11) the sequence $\{\tilde{\mathbf{x}}^{(n)} - \tilde{\mathbf{x}}^{(n+1)}\}$ converges to 0.

For the discussion to follow, consider the surrogate function for the PML objective function Φ at the iterate $\tilde{\mathbf{x}}^{(n)}$, $\tilde{\Phi}^{(n)}$, (see (3.29))

$$\tilde{\Phi}^{(n)}(\mathbf{x}) \triangleq \sum_{j=1}^J \tilde{\phi}_j^{(n)}(x_j) + \tilde{C}_3^{(n)}, \quad (4.1)$$

where $\tilde{\phi}_j^{(n)}(t) \triangleq \tilde{E}_j^{(n)} \log(t) + \tilde{F}_j^{(n)} t^2 + \tilde{G}_j^{(n)} t$ for $t > 0$, $\tilde{C}_3^{(n)}$ is independent of \mathbf{x} , and

$$\tilde{E}_j^{(n)} \triangleq - \sum_{i=1}^I d_i \frac{\mathcal{P}_{ij} \tilde{x}_j^{(n)}}{[\mathcal{P}\tilde{\mathbf{x}}^{(n)} + \boldsymbol{\rho}]}, \quad (4.2)$$

$$\tilde{F}_j^{(n)} \triangleq 2\beta \sum_{k \in N_j} \omega_{jk} \gamma (\tilde{x}_j^{(n)} - \tilde{x}_k^{(n)}) \quad (4.3)$$

$$\tilde{G}_j^{(n)} \triangleq \sum_{i=1}^I \mathcal{P}_{ij} - 2\beta \sum_{k \in N_j} \omega_{jk} \gamma (\tilde{x}_j^{(n)} - \tilde{x}_k^{(n)}) (\tilde{x}_j^{(n)} + \tilde{x}_k^{(n)}) \quad (4.4)$$

(note: $\tilde{\Phi}^{(n)}$, $\tilde{E}_j^{(n)}$, $\tilde{F}_j^{(n)}$, $\tilde{G}_j^{(n)}$, and $\tilde{C}_3^{(n)}$ result by substituting $\tilde{\mathbf{x}}$ for \mathbf{x} in (3.29), (3.30), (3.31), (3.32), and (3.33), respectively). To prove that the whole sequence $\{\chi^{(n)}\}$ converges, as done in the convergence proof of the PML algorithm, we first present the following proposition:

Proposition 5 *Let \mathbf{x}^* be a limit point of the sequence $\{\tilde{\mathbf{x}}^{(n)}\}$. Then, for all j such that $x_j^* \neq 0$,*

$$\frac{\partial}{\partial x_j} \Phi(\mathbf{x}) \Big|_{\mathbf{x}=\mathbf{x}^*} = 0. \quad (4.5)$$

Proof: By (P8), the sequence $\{\tilde{\mathbf{x}}^{(n)}\}$ is bounded and there is a subsequence $\{\tilde{\mathbf{x}}^{(n_l)}\}$ that converges to \mathbf{x}^* . By (P10), the subsequence $\{\mathbf{x}^{(n_l+1)}\}$ also converges to \mathbf{x}^* . Recall that $\dot{\tilde{\phi}}_j^{(n_l)}(t) = \tilde{E}_j^{(n_l)}/t + 2\tilde{F}_j^{(n_l)}t + \tilde{G}_j^{(n_l)}$. Now, if $x_j^* \neq 0$, it follows that $\tilde{E}_j^{(n_l)}/\tilde{x}_j^{(n_l)}$ and $\tilde{E}_j^{(n_l)}/x_j^{(n_l+1)}$ converge to the same limit, which in turn implies that

$$\lim_{l \rightarrow \infty} \dot{\tilde{\phi}}_j^{(n_l)}(\tilde{x}_j^{(n_l)}) = \lim_{l \rightarrow \infty} \dot{\tilde{\phi}}_j^{(n_l)}(x_j^{(n_l+1)}). \quad (4.6)$$

Since $\dot{\tilde{\phi}}_j^{(n_l)}(x_j^{(n_l+1)}) = 0$ by (3.36), and $\nabla \tilde{\Phi}^{(n_l)}(\tilde{\mathbf{x}}^{(n_l)}) = \nabla \Phi(\tilde{\mathbf{x}}^{(n_l)})$, it can be said that

$$\frac{\partial}{\partial x_j} \Phi(\mathbf{x}) \Big|_{\mathbf{x}=\mathbf{x}^*} = 0 \quad \text{for all } j \text{ such that } x_j^* \neq 0. \quad (4.7)$$

Theorem 2 *The sequence $\{\chi^{(n)}\}$ converges to the unique minimizer of Φ .*

Proof: Let \mathbf{x}^* be a limit point of the sequence $\{\tilde{\mathbf{x}}^{(n)}\}$. Since the set of limit points of the sequence $\{\tilde{\mathbf{x}}^{(n)}\}$ is connected by (P8) and (P11) (see [60, p. 173] and *Theorem 1* in Chapter 3) and there are a finite number of limit points of $\{\tilde{\mathbf{x}}^{(n)}\}$ by *Proposition 4* and *Proposition 5*, it follows that there is only one limit point in the set of limit points. Thus, $\{\tilde{\mathbf{x}}^{(n)}\} \rightarrow \mathbf{x}^*$. Since $\lim\{\|\tilde{\mathbf{x}}^{(n)} - \mathbf{x}^*\|^2\} = 0$, by (P10) we have $\lim\{\|\mathbf{x}^{(n+1)} - \mathbf{x}^*\|^2\} = 0$ (note: $\|\mathbf{x}^{(n+1)} - \mathbf{x}^*\|^2 \leq \|\mathbf{x}^{(n+1)} - \tilde{\mathbf{x}}^{(n)}\|^2 + \|\tilde{\mathbf{x}}^{(n)} - \mathbf{x}^*\|^2$). Hence, $\{\mathbf{x}^{(n+1)}\} \rightarrow \mathbf{x}^*$ (i.e., $\{\mathbf{x}^{(n)}\} \rightarrow \mathbf{x}^*$). Therefore we can deduce that the whole sequence $\{\chi^{(n)}\} \rightarrow \mathbf{x}^*$. To prove the sequence $\{\chi^{(n)}\}$ converges to the unique minimizer of the PML objective function, we must show that \mathbf{x}^* satisfies the Kuhn-Tucker conditions (i.e., (3.48), (3.49), and (3.50)). Since all the points in the sequence $\{\chi^{(n)}\}$ are positive, it must be true that the limit point \mathbf{x}^* is nonnegative (i.e., (3.48) is satisfied). By *Proposition 5*,

$$\frac{\partial}{\partial x_j} \Phi(\mathbf{x}) \Big|_{\mathbf{x}=\mathbf{x}^*} = 0 \quad \text{for all } j \text{ such that } x_j^* \neq 0. \quad (4.8)$$

Thus, (3.49) is satisfied. For j such that $x_j^* = 0$, suppose

$$\frac{\partial}{\partial x_j} \Phi(\mathbf{x}) \Big|_{\mathbf{x}=\mathbf{x}^*} < 0. \quad (4.9)$$

Then, it follows that $\lim_{n \rightarrow \infty} \dot{\phi}_j^{(n)}(\tilde{x}_j^{(n)}) < 0$ by the property $\nabla \tilde{\Phi}^{(n)}(\tilde{\mathbf{x}}^{(n)}) = \nabla \Phi(\tilde{\mathbf{x}}^{(n)})$, and $\dot{\phi}_j^{(n)}(\tilde{x}_j^{(n)}) < 0$ for sufficiently large n . Consider $\dot{\phi}_j^{(n)}(x_j^{(n+1)}) = \tilde{E}_j^{(n)} / x_j^{(n+1)} + 2\tilde{F}_j^{(n)}x_j^{(n+1)} + \tilde{G}_j^{(n)}$. If $x_j^{(n+1)} < \tilde{x}_j^{(n)}$, then

$$\frac{x_j^{(n+1)}}{\tilde{x}_j^{(n)}} \cdot \frac{\tilde{E}_j^{(n)}}{x_j^{(n+1)}} + 2\tilde{F}_j^{(n)}x_j^{(n+1)} + \tilde{G}_j^{(n)} > 0 \quad (4.10)$$

because $\dot{\phi}_j^{(n)}(x_j^{(n+1)}) = 0$ by (3.36) and $\tilde{E}_j^{(n)}/x_j^{(n+1)} < 0$. Moreover, the fact that \tilde{F}_j is positive implies that

$$\frac{\tilde{E}_j^{(n)}}{\tilde{x}_j^{(n)}} + 2\tilde{F}_j^{(n)}\tilde{x}_j^{(n)} + \tilde{G}_j^{(n)} = \dot{q}_j^{(n)}(\tilde{x}_j^{(n)}) > 0 , \quad (4.11)$$

which is a contradiction. Thus, $x_j^{(n+1)} \geq \tilde{x}_j^{(n)}$ for all sufficiently large n . However, this contradicts the fact $\chi_j^{(n)} \rightarrow 0$. Therefore,

$$\frac{\partial}{\partial x_j} \Phi(\mathbf{x}) \Big|_{\mathbf{x}=\mathbf{x}^*} \geq 0 \quad \text{for all } j \text{ such that } x_j^* = 0 . \quad (4.12)$$

This satisfies (3.50). ■

4.2 Accelerated Penalized Maximum Likelihood (APML) Algorithm

In Section 4.1, we showed that the sequence $\{\boldsymbol{\chi}^{(n)}\} = \{\tilde{\mathbf{x}}^{(0)}, \mathbf{x}^{(1)}, \tilde{\mathbf{x}}^{(1)}, \mathbf{x}^{(2)}, \tilde{\mathbf{x}}^{(2)}, \dots\}$ converges to the nonnegative minimizer of the PML objective function Φ if: (1) $\tilde{\mathbf{x}}^{(0)} > \mathbf{0}$, (2) $\mathbf{x}^{(n+1)}$ is the nonnegative minimizer of the surrogate function for the PML objective function Φ at the iterate $\tilde{\mathbf{x}}^{(n)}$, $\tilde{\Phi}^{(n)}$, and (3) $\tilde{\mathbf{x}}^{(n)} > \mathbf{0}$ satisfies (C4) and (C5). In this section, we present an algorithm that produces such a sequence $\{\boldsymbol{\chi}^{(n)}\}$.

First, consider the following steps: given an initial guess $\tilde{\mathbf{x}}^{(0)} > \mathbf{0}$, for $n = 0, 1, 2, \dots$

- **Step 1** Get the standard PML iterate $\mathbf{x}^{(n+1)} \triangleq \arg \min_{\mathbf{x}} \tilde{\Phi}^{(n)}(\mathbf{x})$ subject to $\mathbf{x} \geq \mathbf{0}$.
- **Step 2** Get the accelerated PML iterate $\tilde{\mathbf{x}}^{(n+1)} \triangleq \mathbf{x}^{(n+1)} + \alpha_*^{(n+1)} \mathbf{v}^{(n+1)}$, where $\mathbf{v}^{(n+1)} \neq \mathbf{0}$ is the chosen search direction (i.e., direction vector) and

$$\alpha_*^{(n+1)} \triangleq \arg \min_{\alpha} \Phi(\mathbf{x}^{(n+1)} + \alpha \mathbf{v}^{(n+1)}) . \quad (4.13)$$

- **Step 3** Repeat the steps above until some chosen stopping criterion is met.

With $\mathbf{v}^{(n+1)} = \mathbf{x}^{(n+1)} - \tilde{\mathbf{x}}^{(n)}$, **Step 2** is the pattern search step put forth by Hooke and Jeeve [41, pp. 287-291].

We now modify **Step 2** so that the sequence produced by **Steps 1**, **2**, and **3** converges to the nonnegative minimizer of the PML objective function. **Step 2** does not guarantee that the accelerated iterate is positive. Consequently, we modify the optimal step size as follows

$$\alpha_*^{(n+1)} = \arg \min_{\alpha} \Phi(\mathbf{x}^{(n+1)} + \alpha \mathbf{v}^{(n+1)}) \quad \text{subject to } \alpha \in \mathbb{A}^{(n+1)}, \quad (4.14)$$

where

$$\mathbb{A}^{(n+1)} \triangleq \{\alpha : x_j^{(n+1)} + \alpha v_j^{(n+1)} \geq \zeta^{(n+1)} \text{ for all } j\} \quad (4.15)$$

$$\zeta^{(n+1)} \triangleq \min_j \frac{x_j^{(n+1)}}{n+2} \quad (4.16)$$

Observe, for simplicity, we use the same notation for the optimal step sizes defined by (4.13) and (4.14). For $\alpha \in \mathbb{A}^{(n+1)}$, it is straightforward to see that $\mathbf{x}^{(n+1)} + \alpha \mathbf{v}^{(n+1)} > \mathbf{0}$ because $x_j^{(n+1)} > 0$ for all j and n . Thus, by adding the constraint $\alpha \in \mathbb{A}^{(n+1)}$ to the problem in (4.13), it follows that $\tilde{\mathbf{x}}^{(n)} > 0$ for all n . Because $x_j^{(n+1)} > \zeta^{(n+1)}$ for all j and n , it is evident that $\zeta^{(n+1)}$ has been chosen in such a way that $0 \in \mathbb{A}^{(n+1)}$ for all n . The fact that $0 \in \mathbb{A}^{(n+1)}$ will be used later to prove that the proposed algorithm monotonically decreases the PML objective function Φ with increasing iterations.

Remark : If we allow some elements of $\tilde{\mathbf{x}}^{(n)}$ to be zero, then the feasible region of the function $\Phi(\mathbf{x}^{(n+1)} + \alpha \mathbf{v}^{(n+1)})$ is $\{\alpha : x_j^{(n+1)} + \alpha v_j^{(n+1)} \geq 0 \text{ for all } j\}$. However, for the sequence $\{\mathbf{x}^{(n)}\}$ to converge, we must constrain all elements of $\tilde{\mathbf{x}}^{(n)}$ to be positive because the surrogate function for the PML objective function Φ , $\tilde{\Phi}^{(n)}$, is not defined at $\mathbf{x} = \tilde{\mathbf{x}}^{(n)}$ where $\tilde{x}_j^{(n)} = 0$ for some j . A feasible region of the function $\Phi(\mathbf{x}^{(n+1)} + \alpha \mathbf{v}^{(n+1)})$ that appears natural is

$$\mathbb{A}_0^{(n+1)} \triangleq \{\alpha : x_j^{(n+1)} + \alpha v_j^{(n+1)} > 0 \text{ for all } j\}. \quad (4.17)$$

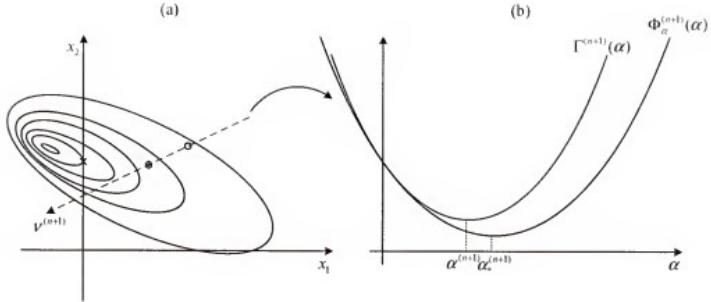


Figure 4: Illustration of the pattern search step: (a) two-dimensional illustration of Φ , (b) one-dimensional slice of the function along the chosen direction vector. The single circle and double circle denote an accelerated iterate $\tilde{\mathbf{x}}^{(n)}$ and a PML iterate $\mathbf{x}^{(n+1)}$, respectively. The mark x denotes the minimizer of the function subject to the constraints $x_1 \geq 0$ and $x_2 \geq 0$.

However, the set $\mathbb{A}_0^{(n+1)}$ is an open set. Consequently, the optimization problem

$$\arg \min_{\alpha} \Phi(\mathbf{x}^{(n+1)} + \alpha \mathbf{v}^{(n+1)}) \quad \text{subject to } \alpha \in \mathbb{A}_0^{(n+1)} \quad (4.18)$$

may not have a solution. To see this fact, suppose that the function $\Phi(\mathbf{x}^{(n+1)} + \alpha \mathbf{v}^{(n+1)})$ is strictly convex and $\mathbb{A}_0^{(n+1)} = \{\alpha : L_0^{(n+1)} < \alpha < U_0^{(n+1)}\}$ for some $L_0^{(n+1)}$ and $U_0^{(n+1)}$. If the function $\Phi(\mathbf{x}^{(n+1)} + \alpha \mathbf{v}^{(n+1)})$ has as its minimizer $\alpha = L_0^{(n+1)}$ over the set $\{\alpha : L_0^{(n+1)} \leq \alpha \leq U_0^{(n+1)}\}$, then the optimization problem in (4.18) does not have a solution.

One more point to mention about the set $\mathbb{A}^{(n+1)}$ is that, strictly speaking, $\zeta^{(n+1)}$ can be any positive number as long as it is chosen in such a way that $0 \in \mathbb{A}^{(n+1)}$ for all n . However, in addition to the condition $0 \in \mathbb{A}^{(n+1)}$, it is reasonable to keep the set $\mathbb{A}^{(n+1)}$ as close as possible to $\mathbb{A}_0^{(n+1)}$ in (4.17). In this sense, the chosen $\zeta^{(n+1)}$ in (4.16) is optimal when n is sufficiently large. ■

Another issue with **Step 2** is that it is hard to solve the optimization problem in (4.13). A sub-optimal solution can be found by determining a surrogate function

for $\Phi_\alpha^{(n+1)}(\alpha) \stackrel{\triangle}{=} \Phi(\mathbf{x}^{(n+1)} + \alpha \mathbf{v}^{(n+1)})$, which we denote by $\Gamma^{(n+1)}(\alpha)$, that satisfies the following conditions:

- (C6) $\Gamma^{(n+1)}(\alpha) \geq \Phi_\alpha^{(n+1)}(\alpha)$ for $\alpha \in \mathbb{A}^{(n+1)}$
- (C7) $\Gamma^{(n+1)}(\alpha) = \Phi_\alpha^{(n+1)}(\alpha)$ for $\alpha = 0$.

By incorporating the surrogate function $\Gamma^{(n+1)}$ with the constraint $\alpha \in \mathbb{A}^{(n+1)}$, an alternative to **Step 2** is:

- **Step 2a** Get the accelerated PML iterate $\tilde{\mathbf{x}}^{(n+1)} \stackrel{\triangle}{=} \mathbf{x}^{(n+1)} + \alpha^{(n+1)} \mathbf{v}^{(n+1)}$, where

$$\alpha^{(n+1)} \stackrel{\triangle}{=} \arg \min_{\alpha} \Gamma^{(n+1)}(\alpha) \quad \text{subject to } \alpha \in \mathbb{A}^{(n+1)}. \quad (4.19)$$

It is important to point out that, for convenience, $\tilde{\mathbf{x}}^{(n+1)}$ has been re-defined in **Step 2a**. This new definition will be used throughout the rest of the dissertation.

In Figure 4-2, the alternative pattern search step with the surrogate function $\Gamma^{(n+1)}$ is illustrated. In Figure 4-2 (a), a two-dimensional example of Φ is shown with a direction vector $\mathbf{v}^{(n+1)}$. In addition, the one-dimensional slice of Φ along the direction vector $\mathbf{v}^{(n+1)}$, which we denote $\Phi_\alpha^{(n+1)}$, and a surrogate function $\Gamma^{(n+1)}$ that satisfies (C6) and (C7) are shown in Figure 4-2 (b).

By design, the sequence $\{\mathbf{x}^{(n)}\}$ produced by **Steps 1**, **2a**, and **3** satisfies the monotonicity condition (C4). To see this fact, note that $\Phi(\tilde{\mathbf{x}}^{(n+1)}) = \Phi_\alpha^{(n+1)}(\alpha^{(n+1)})$ by the definition of $\tilde{\mathbf{x}}^{(n+1)}$. By (C6), it follows that $\Phi_\alpha^{(n+1)}(\alpha^{(n+1)}) \leq \Gamma^{(n+1)}(\alpha^{(n+1)})$. Also, from the definition of $\alpha^{(n+1)}$ in (4.19) and the fact that $0 \in \mathbb{A}^{(n+1)}$, we obtain the result $\Gamma^{(n+1)}(\alpha^{(n+1)}) \leq \Gamma^{(n+1)}(0)$. Finally, by (C7), it can be concluded that $\Phi(\tilde{\mathbf{x}}^{(n+1)}) \leq \Gamma^{(n+1)}(0) = \Phi_\alpha^{(n+1)}(0) = \Phi(\mathbf{x}^{(n+1)})$ for all n .

We now present our choice for the surrogate function $\Gamma^{(n+1)}$ that satisfies (C6) and (C7). First, note that the negative log likelihood function $-\Psi$ can be expressed

as

$$-\Psi(\mathbf{x}) = \sum_{i=1}^I \{[\mathcal{P}\mathbf{x}]_i - d_i \log([\mathcal{P}\mathbf{x} + \boldsymbol{\rho}]_i)\} + \sum_{i=1}^I \{\rho_i + \log(d_i!)\} \quad (4.20)$$

$$= \sum_{i=1}^I \psi_i([\mathcal{P}\mathbf{x}]_i) + C_5, \quad (4.21)$$

where $\psi_i(t) \triangleq t - d_i \log(t + \rho_i)$ and $C_5 \triangleq \sum_{i=1}^I \{\rho_i + \log(d_i!)\}$. Suppose a function, $\theta_i^{(n+1)}$, can be found such that

- (C8) $\theta_i^{(n+1)}(\alpha) \geq \psi_i^{(n+1)}(\alpha)$ for $\alpha \in \mathbb{A}^{(n+1)}$
- (C9) $\theta_i^{(n+1)}(\alpha) = \psi_i^{(n+1)}(\alpha)$ for $\alpha = 0$,

where $\psi_i^{(n+1)}(\alpha) \triangleq \psi_i([\mathcal{P}\mathbf{x}^{(n+1)}]_i + \alpha[\mathcal{P}\mathbf{v}^{(n+1)}]_i)$. Then, the function

$$\Theta^{(n+1)}(\alpha) \triangleq \sum_{i=1}^I \theta_i^{(n+1)}(\alpha) + C_5 \quad (4.22)$$

will satisfy the conditions

- (C10) $\Theta^{(n+1)}(\alpha) \geq -\Psi(\mathbf{x}^{(n+1)} + \alpha\mathbf{v}^{(n+1)})$ for $\alpha \in \mathbb{A}^{(n+1)}$
- (C11) $\Theta^{(n+1)}(\alpha) = -\Psi(\mathbf{x}^{(n+1)} + \alpha\mathbf{v}^{(n+1)})$ for $\alpha = 0$.

A function that satisfies (C8) and (C9) is

$$\theta_i^{(n+1)}(\alpha) \triangleq \mu_i^{(n+1)} \frac{\alpha^2}{2} + \dot{\psi}_i^{(n+1)}(0)\alpha + \psi_i^{(n+1)}(0), \quad (4.23)$$

where $\mu_i^{(n+1)} \triangleq \max\{\dot{\psi}_i^{(n+1)}(\alpha) \text{ subject to } \alpha \in \mathbb{A}^{(n+1)}\}$. From the definition of $\theta_i^{(n+1)}$, it is obvious that $\theta_i^{(n+1)}(0) = \psi_i^{(n+1)}(0)$. Thus, (C9) is satisfied. To see that $\theta_i^{(n+1)}$ satisfies (C8), consider the function $z_i^{(n+1)}(\alpha) = \theta_i^{(n+1)}(\alpha) - \psi_i^{(n+1)}(\alpha)$. From the definition of $\mu_i^{(n+1)}$, it is clear that $\dot{z}_i^{(n+1)} \geq 0$. Thus, it follows that $z_i^{(n+1)}$ is a convex function. Moreover, $\alpha = 0$ is a minimizer of $z_i^{(n+1)}$ because $\dot{z}_i^{(n+1)}(0) = 0$ by the definition of $\theta_i^{(n+1)}$. Since $z_i^{(n+1)}(0) = 0$ by (C9), it is straightforward to see that $z_i^{(n+1)}(\alpha) \geq 0$. This result implies that $\theta_i^{(n+1)}(\alpha) \geq \psi_i^{(n+1)}(\alpha)$ for all $\alpha \in \mathbb{A}^{(n+1)}$. So, (C8) is satisfied. To calculate $\mu_i^{(n+1)}$, it is worthwhile to note that the set $\mathbb{A}^{(n+1)}$ can

be written as $\mathbb{A}^{(n+1)} = \{\alpha : L^{(n+1)} \leq \alpha \leq U^{(n+1)}\}$, where

$$L^{(n+1)} \triangleq \max \left\{ \frac{\zeta^{(n+1)} - x_j^{(n+1)}}{v_j^{(n+1)}} : j \text{ such that } v_j^{(n+1)} > 0 \right\} \quad (4.24)$$

$$U^{(n+1)} \triangleq \min \left\{ \frac{\zeta^{(n+1)} - x_j^{(n+1)}}{v_j^{(n+1)}} : j \text{ such that } v_j^{(n+1)} < 0 \right\}. \quad (4.25)$$

Observe that $L^{(n+1)} < 0$ and $U^{(n+1)} > 0$. Since the second derivative of $\psi_i^{(n+1)}$ is

$$\ddot{\psi}_i^{(n+1)}(\alpha) = \frac{d_i([\mathcal{P}\mathbf{v}^{(n+1)}]_i)^2}{([\mathcal{P}\mathbf{v}^{(n+1)}]_i\alpha + [\mathcal{P}\mathbf{x}^{(n+1)}]_i + \rho_i)^2}, \quad (4.26)$$

the maximum second derivative of $\psi_i^{(n+1)}$ for $\alpha \in \mathbb{A}^{(n+1)}$ is

$$\mu_i^{(n+1)} = \begin{cases} \frac{d_i([\mathcal{P}\mathbf{v}^{(n+1)}]_i)^2}{([\mathcal{P}\mathbf{v}^{(n+1)}]_i L^{(n+1)} + [\mathcal{P}\mathbf{x}^{(n+1)}]_i + \rho_i)^2}, & [\mathcal{P}\mathbf{v}^{(n+1)}]_i > 0 \\ \frac{d_i([\mathcal{P}\mathbf{v}^{(n+1)}]_i)^2}{([\mathcal{P}\mathbf{v}^{(n+1)}]_i U^{(n+1)} + [\mathcal{P}\mathbf{x}^{(n+1)}]_i + \rho_i)^2}, & [\mathcal{P}\mathbf{v}^{(n+1)}]_i < 0 \\ 0, & [\mathcal{P}\mathbf{v}^{(n+1)}]_i = 0. \end{cases} \quad (4.27)$$

It should be noted that $[\mathcal{P}\mathbf{v}^{(n+1)}]_i\alpha + [\mathcal{P}\mathbf{x}^{(n+1)}]_i + \rho_i > 0$ for $\alpha \in \mathbb{A}^{(n+1)}$ (see **(AS1)**).

At this point, we need a surrogate function for the penalty function Λ once again. As mentioned in Chapter 3, under assumptions **(AS3)-(AS8)**, Huber developed a surrogate function for λ , denoted by $\bar{\lambda}^{(n)}$ (see (3.14)). By the properties of the surrogate function $\bar{\lambda}^{(n)}$, it is clear that the surrogate function $\bar{\Lambda}^{(n)}$, which is defined by

$$\bar{\Lambda}^{(n)}(\mathbf{x}) = \sum_{j=1}^J \sum_{k \in N_j} \omega_{jk} \bar{\lambda}^{(n)}(x_j - x_k), \quad (4.28)$$

satisfies $\bar{\Lambda}^{(n)}(\mathbf{x}) \geq \Lambda(\mathbf{x})$ for all \mathbf{x} and $\bar{\Lambda}^{(n)}(\mathbf{x}^{(n)}) = \Lambda(\mathbf{x}^{(n)})$. Thus, $\bar{\Lambda}^{(n+1)}(\mathbf{x}^{(n+1)} + \alpha \mathbf{v}^{(n+1)})$ will satisfy

- **(C12)** $\bar{\Lambda}^{(n+1)}(\mathbf{x}^{(n+1)} + \alpha \mathbf{v}^{(n+1)}) \geq \Lambda(\mathbf{x}^{(n+1)} + \alpha \mathbf{v}^{(n+1)})$ for $\alpha \in \mathbb{A}^{(n+1)}$
- **(C13)** $\bar{\Lambda}^{(n+1)}(\mathbf{x}^{(n+1)} + \alpha \mathbf{v}^{(n+1)}) = \Lambda(\mathbf{x}^{(n+1)} + \alpha \mathbf{v}^{(n+1)})$ for $\alpha = 0$.

Finally, by **(C10)-(C13)**, it is clear that the function

$$\Gamma^{(n+1)}(\alpha) \triangleq \Theta^{(n+1)}(\alpha) + \beta \bar{\Lambda}^{(n+1)}(\mathbf{x}^{(n+1)} + \alpha \mathbf{v}^{(n+1)}) \quad (4.29)$$

satisfies (C6) and (C7).

To solve (4.19), we first determine the unconstrained minimizer of $\Gamma^{(n+1)}$:

$$\bar{\alpha}^{(n+1)} \triangleq \arg \min_{\alpha} \Gamma^{(n+1)}(\alpha). \quad (4.30)$$

Since $\Gamma^{(n+1)}$ is strictly convex by $\gamma(t) > 0$ for $-\infty < t < \infty$, $\mathbf{v}^{(n+1)} \neq \mathbf{0}$, (AS1), and (AS2) (see Appendix D), the expression for $\bar{\alpha}^{(n+1)}$ can be found by simply computing the first derivative of $\Gamma^{(n+1)}$ and setting it to zero (see Appendix D):

$$\bar{\alpha}^{(n+1)} = \frac{-\sum_{i=1}^I \dot{\psi}_i^{(n+1)}(0) - \beta \sum_{j=1}^J \sum_{k \in N_j} \omega_{jk} \dot{\lambda}(x_j^{(n+1)} - x_k^{(n+1)}) (v_j^{(n+1)} - v_k^{(n+1)})}{\sum_{i=1}^I \mu_i^{(n+1)} + \beta \sum_{j=1}^J \sum_{k \in N_j} \omega_{jk} \gamma(x_j^{(n+1)} - x_k^{(n+1)}) (v_j^{(n+1)} - v_k^{(n+1)})^2}. \quad (4.31)$$

Given (4.24), (4.25), and (4.31), the solution to the constrained optimization problem in (4.19) is

$$\alpha^{(n+1)} = \begin{cases} U^{(n+1)}, & \text{if } \bar{\alpha}^{(n+1)} \geq U^{(n+1)} \\ L^{(n+1)}, & \text{if } \bar{\alpha}^{(n+1)} \leq L^{(n+1)} \\ \bar{\alpha}^{(n+1)}, & \text{otherwise} \end{cases}. \quad (4.32)$$

All that remains now is to show that the sequence $\{\chi^{(n)}\}$ produced by Step 1, Step 2a with $\Gamma^{(n+1)}$ in (4.29), and Step 3 satisfies (C5). To see this, note that by (C6), (C7), and the definition of $\tilde{\mathbf{x}}^{(n+1)}$, we have the following inequality

$$\Phi(\mathbf{x}^{(n+1)}) - \Phi(\tilde{\mathbf{x}}^{(n+1)}) \geq \Gamma^{(n+1)}(0) - \Gamma^{(n+1)}(\alpha^{(n+1)}). \quad (4.33)$$

Suppose, for each n , the function $\Gamma^{(n+1)}$ is expanded into a second-order Taylor series about the point $\alpha^{(n+1)}$ and evaluated at $\alpha = 0$. Then, the right hand side of (4.33) can be written as

$$\Gamma^{(n+1)}(0) - \Gamma^{(n+1)}(\alpha^{(n+1)}) = \dot{\Gamma}^{(n+1)}(\alpha^{(n+1)})(-\alpha^{(n+1)}) + \frac{1}{2} \ddot{\Gamma}^{(n+1)}(\tilde{\alpha}^{(n+1)})(-\alpha^{(n+1)})^2, \quad (4.34)$$

where $\tilde{\alpha}^{(n+1)}$ is a point between 0 and $\alpha^{(n+1)}$. By the strict convexity of $\Gamma^{(n+1)}$, $L^{(n+1)} < 0$, and $U^{(n+1)} > 0$, it follows that $\ddot{\Gamma}^{(n+1)}(\alpha^{(n+1)})(-\alpha^{(n+1)}) \geq 0$ for all n . Thus,

$$\Gamma^{(n+1)}(0) - \Gamma^{(n+1)}(\alpha^{(n+1)}) \geq \frac{1}{2}\ddot{\Gamma}^{(n+1)}(\tilde{\alpha}^{(n+1)})(\alpha^{(n+1)})^2. \quad (4.35)$$

Since there exists a symmetric positive definite matrix \mathcal{M} , which is independent of n , such that $\ddot{\Gamma}^{(n+1)}(\tilde{\alpha}^{(n+1)}) \geq 2(\mathbf{v}^{(n+1)})'\mathcal{M}(\mathbf{v}^{(n+1)})$ (see Appendix E), it follows that

$$\Gamma^{(n+1)}(0) - \Gamma^{(n+1)}(\alpha^{(n+1)}) \geq (\mathbf{v}^{(n+1)})'\mathcal{M}(\mathbf{v}^{(n+1)})(\alpha^{(n+1)})^2. \quad (4.36)$$

Hence, to prove (C5), we show that there exists some constant $c_2 > 0$ such that

$$(\mathbf{v}^{(n+1)})'\mathcal{M}(\mathbf{v}^{(n+1)}) \geq c_2\|\mathbf{v}^{(n+1)}\|^2, \quad (4.37)$$

where we used the fact that $\|\mathbf{x}^{(n+1)} - \tilde{\mathbf{x}}^{(n+1)}\|^2 = (\alpha^{(n+1)})^2\|\mathbf{v}^{(n+1)}\|^2$ by the definition of $\tilde{\mathbf{x}}^{(n+1)}$. Since \mathcal{M} is a symmetric matrix, it can be factored as $\mathcal{M} = \Upsilon\Delta\Upsilon'$ by the Spectral theorem [63, p.309], where the columns of the matrix Υ contain orthonormal eigenvectors of \mathcal{M} and the diagonal matrix Δ contains corresponding eigenvalues along its diagonal. Using the fact that $\Upsilon'\Upsilon$ produces the $J \times J$ identity matrix and Rayleigh's quotient with $\mathbf{v}^{(n+1)} = \Upsilon\mathbf{z}^{(n+1)}$ (i.e., $\mathbf{z}^{(n+1)} \triangleq \Upsilon'\mathbf{v}^{(n+1)}$) [63, p.348], it follows that

$$\frac{(\mathbf{v}^{(n+1)})'\mathcal{M}(\mathbf{v}^{(n+1)})}{\|\mathbf{v}^{(n+1)}\|^2} = \frac{(\Upsilon\mathbf{z}^{(n+1)})'\mathcal{M}(\Upsilon\mathbf{z}^{(n+1)})}{(\Upsilon\mathbf{z}^{(n+1)})'(\Upsilon\mathbf{z}^{(n+1)})} \quad (4.38)$$

$$= \frac{(\mathbf{z}^{(n+1)})'\Delta(\mathbf{z}^{(n+1)})}{(\mathbf{z}^{(n+1)})'(\mathbf{z}^{(n+1)})} \quad (4.39)$$

$$= \frac{\sum_j e_j(z_j^{(n+1)})^2}{\sum_j(z_j^{(n+1)})^2} \quad (4.40)$$

$$\geq e_m, \quad (4.41)$$

where $\{e_j\}_{j=1}^J$ are the eigenvalues of \mathcal{M} and e_m is the smallest eigenvalue. Since \mathcal{M} is positive definite, e_m is positive. Therefore, with $c_2 = e_m > 0$, we obtain

$$(\mathbf{v}^{(n+1)})' \mathcal{M}(\mathbf{v}^{(n+1)}) \geq c_2 \|\mathbf{v}^{(n+1)}\|^2 \text{ and}$$

$$\Phi(\mathbf{x}^{(n+1)}) - \Phi(\tilde{\mathbf{x}}^{(n+1)}) \geq c_2 \|\mathbf{x}^{(n+1)} - \tilde{\mathbf{x}}^{(n+1)}\|^2. \quad (4.42)$$

4.3 Direction Vectors

In some algorithms, the gradient of a function is used as a direction vector, as in the method of steepest descent [55, ch. 14]. Suppose the direction vector $\mathbf{v}^{(n+1)}$ was chosen to be the gradient of the PML objective function Φ evaluated at the PML iterate $\mathbf{x}^{(n+1)}$ (i.e., $\mathbf{v}^{(n+1)} = \nabla \Phi(\mathbf{x}^{(n+1)})$). Then, the gradient must be calculated at each step. However, the computational cost of the gradient of Φ is on the same order as the computational cost of single PML iteration. Moreover, in experiments, the APML algorithm with $\mathbf{v}^{(n+1)} = \nabla \Phi(\mathbf{x}^{(n+1)})$ decreased the PML objective function slower than the PML algorithm.

The current direction vector in the pattern search step by Hooke and Jeeve is the difference of the two most recent iterations. Specifically, the direction vector is defined by $\tilde{\mathbf{v}}^{(n+1)} \triangleq \mathbf{x}^{(n+1)} - \tilde{\mathbf{x}}^{(n)}$. This choice can be justified in a reasonable manner. To see why, assume that a closed-form expression is available for the minimizer of $\Phi(\mathbf{x}^{(n+1)} + \alpha \mathbf{v}^{(n+1)})$. Then, the “best” direction vector is $(\mathbf{x}^{(n+1)} - \mathbf{x}^*)$ because $\Phi(\mathbf{x}^{(n+1)} + \alpha \mathbf{v}^{(n+1)})$ “contains” the point \mathbf{x}^* (note: \mathbf{x}^* would result with $\alpha = -1$), where \mathbf{x}^* is the minimizer of the PML objective function. However, \mathbf{x}^* is not known at the n^{th} iteration. The “best” estimate of \mathbf{x}^* at the n^{th} iteration is $\tilde{\mathbf{x}}^{(n)}$, namely the accelerated iterate at the $(n - 1)^{th}$ iteration. In this section, we introduce a direction vector that works better than Hook and Jeeve’s direction vector in terms of convergence rate.

The direction vector we choose is a simple variation of $\tilde{\mathbf{v}}^{(n+1)}$ that is not computationally expensive (note: J subtractions are required for $\tilde{\mathbf{v}}^{(n+1)}$). Since there are no positrons emitted outside the subject being scanned, the PML estimate will contain many values near zero. This claim is supported by Figure 4-3. In the figure,

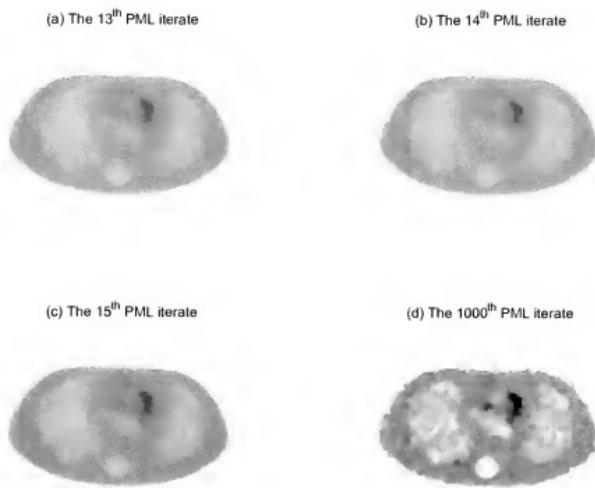


Figure 4-3: PML iterates are shown: (a) the 13th PML iterate, (b) the 14th PML iterate, (c) the 15th PML iterate, and (d) the 1000th PML iterate. The images were generated using a real thorax phantom data. The plane considered contains activity due to the heart, lungs, spine, and background. For display purposes, all the images were adjusted so that they have the same dynamic range.

PML images corresponding to different iteration numbers are shown. The images were generated by applying the PML algorithm to real thorax phantom data (scan duration was 14 minutes) with an uniform initial estimate. The penalty parameter was $\beta = 0.02$ and $\lambda(t) = \log(\cosh(\frac{t}{\delta}))$ with $\delta = 50$ was the penalty function. As can be seen in Figure 4 3 (a), (b), and (c), the early iterates contain values near zero outside of the body. Consider the figure in Figure 4 3 (d), which is the 1000th PML iterate (practically speaking, the iterate is the minimizer of Φ because it did not decrease after the 791st iteration up to the 5000th iteration). The 1000th iterate also contains many values near zero outside the body. Inside the body, on the other hand, the 1000th iterate is very different from the early iterates. From the above observations, it can be said that convergence rate varies significantly between voxels inside the body and voxels outside the body.

From the above observations that the voxels outside the body converge faster than the voxels inside the body and the voxels outside the body converge to values near zero that is the boundary of the set $\{\mathbf{x} : \mathbf{x} \geq \mathbf{0}\}$, we claim that it is better to search for an accelerated iterate along the boundary whenever the current iterate is “near” the boundary. By “boundary”, we mean the set $\{\mathbf{x} \geq \mathbf{0} : x_j = 0 \text{ for some } j\}$. This can be explained by the example shown in Figure 4 4. In Figure 4 4 (a), the second PML iterate is heading toward the x_2 -axis. If we perform the acceleration step with the direction vector $\bar{\nu}^{(n+1)}$, then the accelerated iterate will lie on the x_2 -axis as shown in Figure 4 4 (b). A better direction to be searched is the one that is parallel to the x_2 -axis as shown in Figure 4 4 (c). The principle of the proposed direction vector is to exclude the coordinates corresponding to the voxel values that are “near” the boundary. An easy way to incorporate this idea is to determine the voxels whose values are less than a small positive value, ϵ . Specifically, the proposed direction,

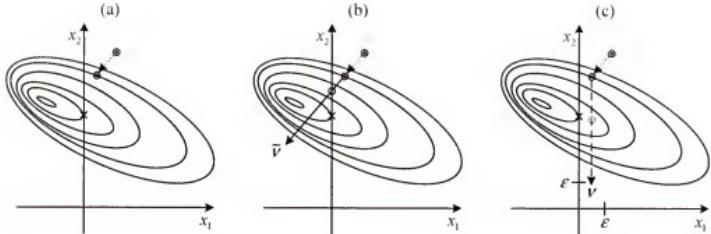


Figure 4: Direction vectors: (a) Standard PML iterates are shown, (b) an accelerated iterate (the single solid circle) is shown by using the direction vector $\bar{\nu}^{(n+1)}$, and (c) an accelerated iterate (the single dotted circle) is shown by using the proposed direction vector $\nu^{(n+1)}$. The double circles denote PML iterates. The mark x denotes the minimizer of the function subject to the constraints $x_1 \geq 0$ and $x_2 \geq 0$.

$\nu^{(n+1)}$ is

$$\nu_j^{(n+1)} \triangleq \begin{cases} 0, & \text{if } x_j^{(n+1)} < \epsilon \\ x_j^{(n+1)} - \bar{x}_j^{(n)}, & \text{otherwise} \end{cases}, \quad (4.43)$$

where $\epsilon \geq 0$ is a user-defined parameter (note: $\nu^{(n+1)} = \bar{\nu}^{(n+1)}$ if $\epsilon = 0$) and $j = 1, 2, \dots, J$.

4.4 Properties of the APML Algorithm

We now provide a summary of the desirable properties of the APML algorithm:

- The APML algorithm needs only one additional parameter (i.e., ϵ), whereas ordered-subsets based algorithms [33,35] need at least three extra parameters (i.e., a relaxation parameter, the number of subsets, and a small positive number that is set to any nonpositive elements of an iterate).
- In experiments, the proposed direction vector performed better than the direction vector put forth by Hooke and Jeeve [41, pp. 287-291].
- The APML algorithm monotonically decreases the PML objective function unlike the algorithms in [23,35].

- The APML algorithm theoretically guarantees nonnegative iterates, whereas some algorithms [33, 35] set any negative element of the iterates to a small positive number.
- The APML algorithm can incorporate a large class of edge-preserving penalty functions unlike the algorithm by De Pierro [30].
- The APML algorithm converges to the minimizer of the PML objective function.

Convergence proofs for the algorithms in [23, 33] are not available.

CHAPTER 5

QUADRATIC EDGE PRESERVING ALGORITHM

In this chapter, we present a regularized image reconstruction algorithm that aims to preserve edges in the reconstructed images so that fine details are more resolvable. We refer to the proposed algorithm as the quadratic edge preserving (QEP) algorithm. The QEP algorithm results via a certain modification of the surrogate function $\Phi^{(n)}$ for the PML objective function. It should be mentioned that the QEP algorithm was first introduced in [18].

Recall that the n^{th} surrogate function for the PML objective function Φ , $\Phi^{(n)}$, can be expressed as $\Phi^{(n)}(\mathbf{x}) = \sum_j \phi_j^{(n)}(x_j) + C_3^{(n)}$ (see (3.29) and (3.35)). For the discussion to follow, it will be convenient to express $\phi_j^{(n)}$ in (3.35) in a different manner

$$\phi_j^{(n)}(t) = E_j^{(n)} \log(t) + F_j^{(n)} t^2 + G_j^{(n)} t \quad (5.1)$$

$$= E_j^{(n)} \log(t) + 2\beta \sum_{k \in N_j} \omega_{jk} \gamma(x_j^{(n)} - x_k^{(n)}) t^2 \quad (5.2)$$

$$+ \left(\sum_{i=1}^I \mathcal{P}_{ij} - 4\beta \sum_{k \in N_j} \omega_{jk} \gamma(x_j^{(n)} - x_k^{(n)}) m_{jk}^{(n)} \right) t \quad (5.3)$$

$$= E_j^{(n)} \log(t) + t \sum_{i=1}^I \mathcal{P}_{ij} + 2\beta \sum_{k \in N_j} \omega_{jk} \gamma(x_j^{(n)} - x_k^{(n)}) (t^2 - 2m_{jk}^{(n)} t) \quad (5.4)$$

$$= l_j^{(n)}(t) + 2\beta \sum_{k \in N_j} h_{jk}^{(n)}(t) - 2\beta \sum_{k \in N_j} \omega_{jk} \gamma(x_j^{(n)} - x_k^{(n)}) m_{jk}^{(n)2}, \quad (5.5)$$

where

$$l_j^{(n)}(t) \triangleq E_j^{(n)} \log(t) + t \sum_{i=1}^I \mathcal{P}_{ij} \quad (5.6)$$

$$h_{jk}^{(n)}(t) = \omega_{jk} \gamma(x_j^{(n)} - x_k^{(n)}) \left(t - m_{jk}^{(n)} \right)^2 \quad (5.7)$$

and $m_{jk}^{(n)}$, $E_j^{(n)}$, $F_j^{(n)}$, and $G_j^{(n)}$ are defined in (3.24), (3.30), (3.31), and (3.32), respectively. Note that the j^{th} element of the next PML iterate $\mathbf{x}^{(n+1)}$, $x_j^{(n+1)}$, is defined to be the nonnegative minimizer of the function $\phi_j^{(n)}$. The function $h_{jk}^{(n)}$ is quadratic so its aperture, $\omega_{jk}\gamma(x_j^{(n)} - x_k^{(n)})$, and minimizer, $m_{jk}^{(n)}$, are expected to play key roles in the regularization process (for a quadratic function, $f(t) = at^2 + bt + c$, the constant a is called the aperture of the function). To see the role of the function $h_{jk}^{(n)}$ on determining $x_j^{(n+1)}$, suppose $\beta = 0$ in (5.5). Then, $\phi_j^{(n)}(t) = l_j^{(n)}(t)$ and the minimizer of the function $\phi_j^{(n)}(t)$ is the “pure” log likelihood iterate (i.e., minimizer of $l_j^{(n)}$). For $\beta \neq 0$, intuitively speaking, it is evident that the functions $\{h_{jk}^{(n)}\}_{k \in N_j}$ act to bias $x_j^{(n+1)}$ from the pure log likelihood iterate towards the minimizers of $\{h_{jk}^{(n)}\}_{k \in N_j}$ (observe: the last term in (5.5) is independent of \mathbf{x}). The degree of influence that the functions $\{h_{jk}^{(n)}\}_{k \in N_j}$ possess is controlled by their apertures and the penalty parameter β .

To highlight the role of the function γ , we let $\beta = \frac{1}{2}$ and $\omega_{jk} = 1$ for all j and k . In this case, the aperture of $h_{jk}^{(n)}$ equals $\gamma(x_j^{(n)} - x_k^{(n)})$. For the quadratic penalty function (i.e., $\lambda(t) = t^2$), the aperture of $h_{jk}^{(n)}$ is a constant for all j , k , and n . Consequently, for all $k \in N_j$, the function $h_{jk}^{(n)}$ has the same degree of influence regardless of the absolute difference between $x_j^{(n)}$ and $x_k^{(n)}$. Practically speaking, this means that the quadratic penalty will overly smooth edges. To preserve edges, it would be helpful to lessen the degree of influence of the functions $\{h_{jk}^{(n)}\}_{k \in N_j}$ whenever the absolute difference between $x_j^{(n)}$ and $x_k^{(n)}$ is sufficiently large. Figure 5–1 is a plot of γ when $\lambda(t) = \log(\cosh(\frac{t}{\delta}))$ (recall: $\gamma(t) \triangleq \dot{\lambda}(t)/t$ in (AS6)) is used in the penalty function with $\delta = 10, 50, 100, 500$. From the figure, it can be seen that γ becomes very small compared with $\gamma(0)$ for $|t| >> \delta$. This means that, when $|x_j^{(n)} - x_k^{(n)}| >> \delta$, $h_{jk}^{(n)}$ will have a “very small” aperture, and consequently the function $h_{jk}^{(n)}$ will not have much influence on $x_j^{(n+1)}$. Said another way, the log-cosh penalty function helps preserve edges whose “heights” are on the order of δ .

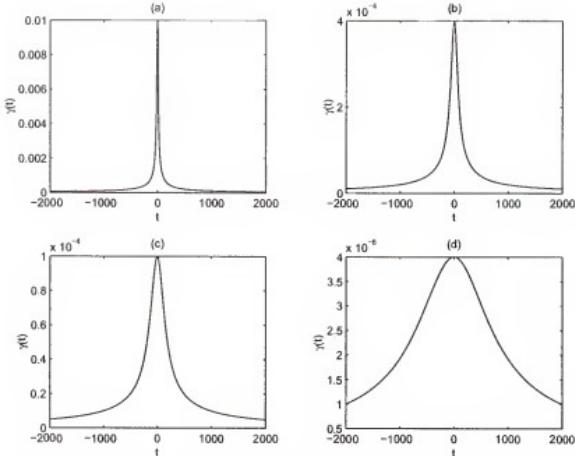


Figure 5.1: The function γ is shown when $\lambda(t) = \log(\cosh(\frac{t}{\delta}))$ is used as the penalty function with: (a) $\delta=10$, (b) $\delta=50$, (c) $\delta=100$, and (d) $\delta=500$.

We will now move the discussion from the aperture of the function $h_{jk}^{(n)}$ to the minimizer of $h_{jk}^{(n)}$. As stated previously, the minimizers of the functions $\{h_{jk}^{(n)}\}_{k \in N_j}$ play an important role in the regularization procedure. More specifically, when the aperture of $h_{jk}^{(n)}$ is sufficiently large, the $(n+1)^{st}$ iterate is biased towards the minimizer of $h_{jk}^{(n)}$, which is $m_{jk}^{(n)} = (x_j^{(n)} + x_k^{(n)})/2$. As a result, there is inherent averaging that takes place with the PML algorithm.

Consider a penalty function, where, for certain functions $\{\sigma_{jk}^{(n)}\}$, it is of the form

$$\Lambda_{\text{ep}}^{(n)}(\mathbf{x}) \triangleq 2 \sum_{j=1}^J \sum_{k \in N_j} \sigma_{jk}^{(n)}(x_j) . \quad (5.8)$$

In order to better preserve edges, we believe an improvement would be to construct $\sigma_{jk}^{(n)}$ so that it has the same aperture as $h_{jk}^{(n)}$, but a different minimizer. The minimizer of $\sigma_{jk}^{(n)}$, which would depend on whether an edge is present, is chosen to be

$$u_{jk}^{(n)} = x_j^{(n)} - \eta(x_j^{(n)} - x_k^{(n)}) , \quad (5.9)$$

where η is a function such that

$$\eta(t) \approx \begin{cases} \xi, & \xi < t \\ -\xi, & t < -\xi \\ t, & \text{otherwise} \end{cases} \quad (5.10)$$

and ξ is a parameter that represents the “heights” of the edges that are to be preserved. An example of a function that can be used is $\eta(t) = \xi \tanh(\frac{t}{\xi})$. In order for $u_{jk}^{(n)}$ to be the minimizer of $\sigma_{jk}^{(n)}$, we define $\sigma_{jk}^{(n)}$ to be

$$\sigma_{jk}^{(n)}(t) = \omega_{jk} \gamma(x_j^{(n)} - x_k^{(n)}) (t - u_{jk}^{(n)})^2. \quad (5.11)$$

Using $\sigma_{jk}^{(n)}$, when the absolute difference between $x_j^{(n)}$ and $x_k^{(n)}$ is less than ξ (i.e., no edge present), smoothing takes place because $u_{jk}^{(n)} \approx x_k^{(n)}$. On the other hand, when the absolute difference between $x_j^{(n)}$ and $x_k^{(n)}$ is greater than ξ (i.e., edge present), less smoothing takes place because there are upper and lower bounds for $u_{jk}^{(n)}$. To help make our idea clearer, consider Figure 5.2. In the figure, $m_{jk}^{(n)}$ and $u_{jk}^{(n)}$ are plotted as a function of $x_k^{(n)}$, where $x_j^{(n)} = 800$ (value is chosen arbitrarily) and $\eta(t) = 250 \tanh(\frac{t}{250})$. It can be observed that $u_{jk}^{(n)}$ is approximately $x_j^{(n)} + 250$ when $x_k^{(n)}$ is larger than $x_j^{(n)} + 250$ and $u_{jk}^{(n)}$ is approximately $x_j^{(n)} - 250$ when $x_k^{(n)}$ is less than $x_j^{(n)} - 250$. In other words, the function η prevents $x_j^{(n+1)}$ from being overly biased towards $x_k^{(n)}$ when the absolute difference between $x_j^{(n)}$ and $x_k^{(n)}$ is greater than 250.

Using the edge preserving penalty function $\Lambda_{\text{ep}}^{(n)}$, an algorithm for obtaining regularized estimates of the emission means follows:

- **Step 1** Let $\mathbf{x}^{(0)} > \mathbf{0}$ be the initial estimate
- **Step 2** Construct $\Lambda_{\text{ep}}^{(n)}(\mathbf{x})$ from the current estimate $\mathbf{x}^{(n)}$ using (5.8), (5.9), (5.10), and (5.11)
- **Step 3** Compute $\mathbf{x}^{(n+1)} = \arg \min \Xi^{(n)}(\mathbf{x})$ subject to $\mathbf{x} \geq \mathbf{0}$, where $\Xi^{(n)}(\mathbf{x}) \triangleq -\Psi(\mathbf{x}) + \beta \Lambda_{\text{ep}}^{(n)}(\mathbf{x})$.
- **Step 4** Iterate between **Steps 2** and **3**.

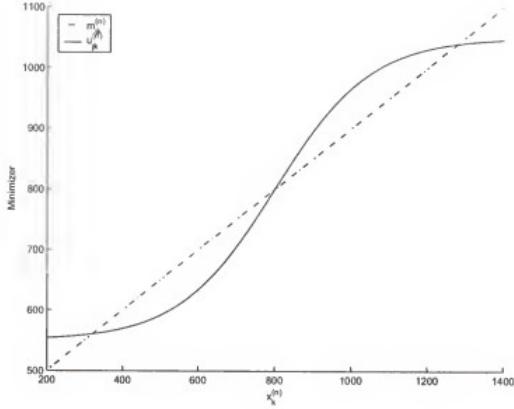


Figure 5.2: Plots of $m_{jk}^{(n)}$ and $u_{jk}^{(n)}$ versus $x_k^{(n)}$ are shown, where $x_j^{(n)} = 800$ is fixed and $\eta(t) = 250 \tanh(\frac{t}{250})$.

It turns out that a closed form expression for the problem in **Step 3** is not possible.

An alternative to **Step 3** is:

- **Step 3a** Find $\mathbf{x}^{(n+1)}$ such that $\Xi^{(n)}(\mathbf{x}^{(n+1)}) \leq \Xi^{(n)}(\mathbf{x}^{(n)})$.

The problem in **Step 3a** can be solved by defining $\mathbf{x}^{(n+1)}$ as

$$\mathbf{x}^{(n+1)} \triangleq \arg \min_{\mathbf{x} \geq \mathbf{0}} \Phi_{\text{cp}}^{(n)}(\mathbf{x}), \quad (5.12)$$

where

$$\Phi_{\text{cp}}^{(n)}(\mathbf{x}) \triangleq -\Psi^{(n)}(\mathbf{x}) + \beta \Lambda_{\text{cp}}^{(n)}(\mathbf{x}) \quad (5.13)$$

and $\Psi^{(n)}$ is defined in (3.12). Defining the iterates according (5.12) and (5.13) insures that $\Xi^{(n)}(\mathbf{x}^{(n+1)}) \leq \Xi^{(n)}(\mathbf{x}^{(n)})$. To see this fact, note that $\Xi^{(n)}(\mathbf{x}^{(n+1)}) \leq \Phi_{\text{cp}}^{(n)}(\mathbf{x}^{(n+1)})$ because $\Psi^{(n)}(\mathbf{x}) \leq \Psi(\mathbf{x})$ for all $\mathbf{x} > \mathbf{0}$. Since $\Phi_{\text{cp}}^{(n)}(\mathbf{x}^{(n+1)}) \leq \Phi_{\text{cp}}^{(n)}(\mathbf{x}^{(n)})$ by (5.12), it follows that

$$\Xi^{(n)}(\mathbf{x}^{(n+1)}) \leq \Phi_{\text{cp}}^{(n)}(\mathbf{x}^{(n+1)}) \leq \Phi_{\text{cp}}^{(n)}(\mathbf{x}^{(n)}) = \Xi^{(n)}(\mathbf{x}^{(n)}). \quad (5.14)$$

All that remains now is to solve the optimization problem in (5.12). Observe that $\Phi_{\text{cp}}^{(n)}$ can be written as

$$\begin{aligned}\Phi_{\text{cp}}^{(n)}(\boldsymbol{x}) &= -\Psi^{(n)}(\boldsymbol{x}) + \beta \Lambda_{\text{cp}}^{(n)}(\boldsymbol{x}) \\ &= \sum_{i=1}^I \left\{ [\mathcal{P}\boldsymbol{x}]_i - \sum_{j=1}^J d_i \frac{\mathcal{P}_{ij}x_j^{(n)}}{[\mathcal{P}\boldsymbol{x}^{(n)} + \boldsymbol{\rho}]_i} \log(x_j) \right\} + C_1^{(n)}\end{aligned}\quad (5.15)$$

$$\begin{aligned}&\quad + 2\beta \sum_{j=1}^J \sum_{k \in N_j} \sigma_{jk}^{(n)}(x_j) \\ &= \sum_{j=1}^J \left\{ x_j \sum_{i=1}^I \mathcal{P}_{ij} - \log(x_j) \sum_{i=1}^I d_i \frac{\mathcal{P}_{ij}x_j^{(n)}}{[\mathcal{P}\boldsymbol{x}^{(n)} + \boldsymbol{\rho}]_i} \right\} + C_1^{(n)} \\ &\quad + \sum_{j=1}^J \left\{ 2\beta \sum_{k \in N_j} \omega_{jk} \gamma(x_j^{(n)} - x_k^{(n)}) (x_j^2 - 2u_{jk}^{(n)}x_j + u_{jk}^{(n)2}) \right\} \quad (5.17)\end{aligned}$$

$$= \sum_{j=1}^J \{ E_j^{(n)} \log(x_j) + F_j^{(n)} x_j^2 + H_j^{(n)} x_j \} + C_4^{(n)}, \quad (5.18)$$

where

$$H_j^{(n)} \triangleq \sum_{i=1}^I \mathcal{P}_{ij} - 4\beta \sum_{k \in N_j} \omega_{jk} \gamma(x_j^{(n)} - x_k^{(n)}) u_{jk}^{(n)} \quad (5.19)$$

$$C_4^{(n)} = C_1^{(n)} + 2\beta \sum_{j=1}^J \sum_{k \in N_j} \omega_{jk} \gamma(x_j^{(n)} - x_k^{(n)}) u_{jk}^{(n)2}, \quad (5.20)$$

and $C_1^{(n)}$, $E_j^{(n)}$, and $F_j^{(n)}$ are defined in (3.13), (3.30), and (3.31), respectively. Since $\Phi_{\text{cp}}^{(n)}$ is de-coupled, it follows that the solution to (5.12) is given by

$$x_j^{(n+1)} = \arg \min_{x_j \geq 0} \{ E_j^{(n)} \log(x_j) + F_j^{(n)} x_j^2 + H_j^{(n)} x_j \}, \quad j = 1, 2, \dots, J. \quad (5.21)$$

Repeating the steps used to derive (3.37), it is straightforward to see that the solution to the optimization problem in (5.21) is

$$x_j^{(n+1)} = \frac{-H_j^{(n)} + \sqrt{H_j^{(n)2} - 8E_j^{(n)}F_j^{(n)}}}{4F_j^{(n)}}, \quad j = 1, 2, \dots, J. \quad (5.22)$$

CHAPTER 6

JOINT EMISSION MEAN AND PROBABILITY MATRIX ESTIMATION

In Chapters 3, 4, and 5, we assumed that the probability matrix \mathcal{P} accounts for errors due to attenuation, detector inefficiency, detector penetration, non-collinearity, and scatter. However, we now assume that an $I \times J$ corrected matrix \mathcal{P}^c is available that accounts for errors due to attenuation, detector inefficiency, and detector penetration. And, we develop a method for correcting errors caused by scatter and non-collinearity. Before presenting the proposed method, we briefly review standard approaches for obtaining \mathcal{P}^c in practice.

The $(i, j)^{th}$ element of \mathcal{P}^c , \mathcal{P}_{ij}^c , denotes the probability that an annihilation in the j^{th} voxel leads to a photon pair being recorded by the i^{th} detector pair when there are no errors due to scatter and non-collinearity. In practice, one can reliably estimate the probability matrix \mathcal{P}^c by using the detector penetration correction method in [7] with an attenuation correction method [9] and detector inefficiency correction method in [15]. However, other correction methods for detector penetration, attenuation, and detector inefficiency could be used in conjunction.

To address attenuation errors, two scans, known as a transmission scan and blank scan, are taken. During a transmission scan, 1 – 3 rotating-rods filled with positron-emitting isotopes rotate outside the subject and the transmission data are the coincidence sums. A blank scan is taken in the same way as the transmission scan except that there is no subject inside the scanner. The coincidence sums during the blank scan form the blank scan data. Given the transmission and blank scan data, an estimate of the attenuation map can be generated. Using the estimated attenuation map, attenuation correction factors are computed and used for attenuation correction. Numerous attenuation estimation methods have been proposed [8–10].

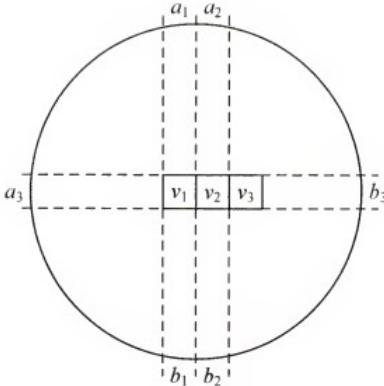


Figure 6.1: A simple example to illustrate the geometry of PET image reconstruction with three voxels (v_1 , v_2 , and v_3) and three detector pairs $((a_1, b_1), (a_2, b_2)$, and (a_3, b_3) . The dashed lines define the tubes of the detector pairs.

To address errors due to detector inefficiency, a blank scan of extremely long duration is performed. From the blank scan, the relative efficiency of the detector pairs can be estimated. Recall that the efficiency of a detector pair is defined to be the probability that a coincidence is recorded when a photon pair is incident on the detectors. Consider a set of detector pairs where each detector pair has same spatial extent. One simple way to estimate the efficiency of a detector pair in the set of detector pairs is to define its efficiency to be the ratio of the number of photon pairs recorded by the detector pair and the mean number of photon pairs recorded by a detector pair in the set of detector pairs. Once, the estimates of efficiencies for all detector pairs are available, they can be incorporated into the probability matrix. To address detector inefficiency, more complicated correction methods such as [13–15] have been proposed.

As discussed in Section 1.3, two photons generated by an annihilation usually do not propagate in exactly opposite directions, which is called non-collinearity of line-of-response [5]. As illustrated in Figure 1–5, it is possible that an annihilation

is recorded by detector pair i_1 given that the annihilation would have been recorded by detector pair i_2 if both photons propagate in exactly opposite directions, where $i_1 \neq i_2$.

Since scatter depends on the activity and attenuation within the subject and the scanner design, it is not straightforward to correct for errors due to scatter. Photons lose energy when they are under gone Compton scattering. Thus, the energy of detected photons may be used to discriminate between unscattered photons and scattered photons [11, pp. 65-69]. Scatter correction methods that are based on multiple (two or more) energy windows have a limitation because detectors have finite energy resolution [46, 47]. As discussed in Chapter 2, state-of-the-art scatter correction methods, such as the method in [52] where the scatter distribution is calculated using an analytical equation, are not practical because of their extremely high computational cost. Consequently, a simple scatter correction method would be beneficial to the PET community. In this chapter, we propose a method that estimates the probability matrix in such a way that errors due to scatter and non-collinearity are addressed.

In Section 6.1, we first propose a model for emission data that accounts for errors due to scatter and non-collinearity. Then, in Section 6.2, we present a method where the unknown emission mean vector and an unknown matrix in the proposed model are jointly estimated. Finally, we propose an algorithm, referred to as probability correction in projection space (PCiPS) algorithm, that estimates the unknown emission mean vector and the unknown matrix in the proposed model.

6.1 Scatter Matrix Model

Consider Figure 6-1 in which a simplified PET scanner consisting of three detector pairs is depicted. For this discussion, we will focus on the voxels v_1 , v_2 , and v_3 shown in Figure 6-1. Let the detector pairs (a_1, b_1) , (a_2, b_2) , and (a_3, b_3) be the first, second, and third detector pairs, respectively. For the scanner in Figure 6-1, we

assume that

$$\mathcal{P}^c = \begin{bmatrix} 0.04 & 0 & 0 \\ 0 & 0.04 & 0 \\ 0.03 & 0.03 & 0.02 \end{bmatrix}. \quad (6.1)$$

Suppose during a scan there are no errors due to scatter and non-collinearity. Then, the mean number of photon pairs recorded by the detector pair (a_1, b_1) is $\mathcal{P}_{11}^c \bar{x}_1$, where \bar{x}_1 is the mean number of positrons emitted from voxel v_1 . The mean number of photon pairs recorded by the detector pairs (a_2, b_2) and (a_3, b_3) are $\mathcal{P}_{22}^c \bar{x}_2$ and $(\mathcal{P}_{31}^c \bar{x}_1 + \mathcal{P}_{32}^c \bar{x}_2 + \mathcal{P}_{33}^c \bar{x}_3)$, respectively, where \bar{x}_2 and \bar{x}_3 are the mean number of positrons emitted from voxels v_2 and v_3 , respectively. As discussed in Section 1.3, a photon's original flight path is altered when it undergoes Compton scattering. Consequently, an annihilation in v_1 that would have been recorded by detector pair (a_1, b_1) if both photons had not undergone Compton scattering may instead be recorded by detector pair (a_2, b_2) when scatter occurs. When we consider non-collinearity, it is also possible that an annihilation in v_1 is recorded by detector pair (a_2, b_2) given that the annihilation would have been recorded by detector pair (a_1, b_1) if both photons propagate in exactly opposite directions.

Let $\mathcal{K}_{i_1 i_2}$ denotes the conditional probability that an annihilation is recorded by detector pair i_1 given that the annihilation would have been recorded by detector pair i_2 if both photons produced by the annihilation had not undergone Compton scattering and both photons propagate in exactly opposite directions, where $i_1 = 1, 2, \dots, I$ and $i_2 = 1, 2, \dots, I$. It is through these unknown probabilities $\{\mathcal{K}_{i_1 i_2}\}$ that we model scatter and non-collinearity. Now, accounting for scatter and non-collinearity, the mean number of photon pairs recorded by detector pair (a_1, b_1) is $\{\mathcal{K}_{11} \mathcal{P}_{11}^c \bar{x}_1 + \mathcal{K}_{12} \mathcal{P}_{22}^c \bar{x}_2 + \mathcal{K}_{13} (\mathcal{P}_{31}^c \bar{x}_1 + \mathcal{P}_{32}^c \bar{x}_2 + \mathcal{P}_{33}^c \bar{x}_3)\}$. Moreover, the mean number of photon pairs recorded by detector pairs (a_2, b_2) and (a_3, b_3) are $\{\mathcal{K}_{21} \mathcal{P}_{11}^c \bar{x}_1 + \mathcal{K}_{22} \mathcal{P}_{22}^c \bar{x}_2 + \mathcal{K}_{23} (\mathcal{P}_{31}^c \bar{x}_1 + \mathcal{P}_{32}^c \bar{x}_2 + \mathcal{P}_{33}^c \bar{x}_3)\}$ and $\{\mathcal{K}_{31} \mathcal{P}_{11}^c \bar{x}_1 + \mathcal{K}_{32} \mathcal{P}_{22}^c \bar{x}_2 + \mathcal{K}_{33} (\mathcal{P}_{31}^c \bar{x}_1 + \mathcal{P}_{32}^c \bar{x}_2 + \mathcal{P}_{33}^c \bar{x}_3)\}$,

respectively. In matrix notation, the mean of the data $E[\mathbf{D}]$ can be expressed as

$$\begin{aligned} E[\mathbf{D}] &= \begin{bmatrix} \mathcal{K}_{11} & \mathcal{K}_{12} & \mathcal{K}_{13} \\ \mathcal{K}_{21} & \mathcal{K}_{22} & \mathcal{K}_{23} \\ \mathcal{K}_{31} & \mathcal{K}_{32} & \mathcal{K}_{33} \end{bmatrix} \begin{bmatrix} 0.04 & 0 & 0 \\ 0 & 0.04 & 0 \\ 0.03 & 0.03 & 0.02 \end{bmatrix} \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \bar{x}_3 \end{bmatrix} \quad (6.2) \\ &= \mathcal{K}\mathcal{P}^c\bar{\mathbf{x}}, \quad (6.3) \end{aligned}$$

where $\bar{\mathbf{x}} = [\bar{x}_1 \bar{x}_2 \bar{x}_3]'$ and the $I \times I$ matrix \mathcal{K} denotes the first matrix on the right-hand side of (6.2). We will refer to \mathcal{K} as the scatter matrix. Since the emission means can be expressed as $\mathcal{K}\mathcal{P}^c\bar{\mathbf{x}}$ in this model, we define the “true” probability matrix as

$$\mathcal{P}^{\text{true}} \triangleq \mathcal{K}\mathcal{P}^c, \quad (6.4)$$

where the $(i, j)^{\text{th}}$ element of $\mathcal{P}^{\text{true}}$, $\mathcal{P}_{ij}^{\text{true}}$, denotes the probability that an annihilation in the j^{th} voxel leads to a photon pair being recorded by the i^{th} detector pair. To our knowledge, the scatter matrix model we propose is not available in literatures. However, similar factorization of $\mathcal{P}^{\text{true}}$ in (6.4) can be found in [28, 53, 64, 65].

With the definition of $\mathcal{P}^{\text{true}}$ in (6.4), the mean number of photon pairs recorded by the i^{th} detector pair can be expressed as

$$\sum_{j=1}^J \mathcal{P}_{ij}^{\text{true}} x_j = \sum_{j=1}^J \sum_{i_2=1}^I \mathcal{K}_{ii_2} \mathcal{P}_{i_2 j}^c x_j \quad (6.5)$$

$$= \sum_{i_2=1}^I \mathcal{K}_{ii_2} \sum_{j=1}^J \mathcal{P}_{i_2 j}^c x_j \quad (6.6)$$

$$= \sum_{i_2=1}^I \mathcal{K}_{ii_2} [\mathcal{P}^c \mathbf{x}]_{i_2}. \quad (6.7)$$

In other words, the mean number of photon pairs recorded by the i^{th} detector pair is a weighted sum of the mean number of photon pairs recorded by all detector pairs when there are no errors due to scatter and non-collinearity. Regarding the matrix \mathcal{K} , two constraints are necessary. Since \mathcal{K} is a probability matrix, it must be true that $0 \leq \mathcal{K}_{ii_2} \leq 1$ for all i_1 and i_2 . Recall that \mathcal{K}_{ii_2} is the conditional probability that

an annihilation is recorded by detector pair i_1 given that the annihilation would have been recorded by detector pair i_2 if both photons produced by the annihilation had not undergone Compton scattering and both photons propagate in exactly opposite directions. Since a photon pair that is recorded by detector pair i_1 can not be recorded by the other detector pairs, we require that

$$\sum_{i_1=1}^I \mathcal{K}_{i_1 i_2} \leq 1 \quad \text{for all } i_2 . \quad (6.8)$$

6.2 Joint Minimum Kullback-Leibler distance Method

We now consider the Poisson model by Shepp and Vardi [16]. In their model, the emission data \mathbf{d} is an observation of a random vector \mathbf{D} that is Poisson distributed with mean $(\mathcal{P}^{\text{true}} \mathbf{x} + \boldsymbol{\rho})$, where \mathbf{x} is the unknown emission mean vector and $\boldsymbol{\rho}$ is the known mean accidental coincidence rate.

Since \mathbf{d} is the ML estimate of the unknown mean $(\mathcal{P}^{\text{true}} \mathbf{x} + \boldsymbol{\rho})$ ¹, it can be said that $\mathbf{d} \approx (\mathcal{P}^{\text{true}} \mathbf{x} + \boldsymbol{\rho})$. Using the definition in (6.4), we can state that

$$\mathbf{d} \approx \mathcal{K} \mathcal{P}^c \mathbf{x} + \boldsymbol{\rho} . \quad (6.9)$$

It is important to point out that there are two unknowns in (6.9): the scatter matrix \mathcal{K} and emission mean vector \mathbf{x} . Thus, given the emission data \mathbf{d} , mean accidental coincidence rate $\boldsymbol{\rho}$, and probability matrix \mathcal{P}^c , the problem of interest is to estimate the emission mean \mathbf{x} and scatter matrix \mathcal{K} . We define the estimate $(\hat{\mathbf{x}}, \hat{\mathcal{K}})$ to be a

¹ If d is an observation of a random vector D that is Poisson distributed with unknown mean λ , then the likelihood function for d is $Pr\{D = d | \lambda\} = \frac{\lambda^d}{d!} e^{-\lambda}$. Since d maximizes the log likelihood function $\log Pr\{D = d | \lambda\}$, d is the ML estimate of λ .

minimizer of the Kullback-Leibler (KL) distance [66] between \mathbf{d} and $(\mathcal{K}\mathcal{P}^c \mathbf{x} + \boldsymbol{\rho})$:

$$\begin{aligned} (\hat{\mathbf{x}}, \hat{\mathcal{K}}) &\stackrel{\triangle}{=} \arg \min_{(\mathbf{x}, \mathcal{K})} \text{KL}(\mathbf{d}, \mathcal{K}\mathcal{P}^c \mathbf{x} + \boldsymbol{\rho}) \quad \text{subject to} \\ \mathbf{x} &\geq \mathbf{0}, \quad \mathcal{K} \geq \mathbf{0}, \quad \text{and} \quad \sum_{i_1=1}^I \mathcal{K}_{i_1 i_2} \leq 1 \quad \text{for all } i_2, \end{aligned} \quad (6.10)$$

where the KL distance between \mathbf{a} and \mathbf{b} is defined by

$$\text{KL}(\mathbf{a}, \mathbf{b}) = \sum_i \left(a_i \log \frac{a_i}{b_i} - a_i + b_i \right) \quad (6.11)$$

and $\mathcal{K} \geq \mathbf{0}$ means that $\mathcal{K}_{i_1 i_2} \geq 0$ for all i_1 and i_2 . The definition of the estimate of $(\mathbf{x}, \mathcal{K})$ is motivated in part by the fact that in [26], Byrne derived Shepp and Vardi's MLEM algorithm [16] by minimizing the KL distance between the emission data \mathbf{d} and $\mathcal{P}^{\text{true}} \mathbf{x}$ using the alternating projection algorithm by Csiszár and Tusnády [67], where $\mathcal{P}^{\text{true}}$ is assumed to be known. It should be mentioned that Byrne derived the MLEM algorithm when $\boldsymbol{\rho} = \mathbf{0}$.

6.3 Probability Correction in Projection Space (PCiPS) Algorithm

Since it is difficult to solve the problem in (6.10), we propose an alternating minimization algorithm where, in a repetitive fashion, one of the two unknowns (i.e., \mathcal{K} and \mathbf{x}) is estimated while the other is fixed. Suppose an initial estimate for the scatter matrix \mathcal{K} (e.g., an $I \times I$ identity matrix), denoted by $\mathcal{K}^{(0)}$, and an initial estimate $\mathbf{x}^{(0)} > \mathbf{0}$ are available. Then, for $n = 0, 1, 2, \dots$, the steps of the proposed algorithm are

- **Step 1** Get the current estimate of emission mean vector $\mathbf{x}^{(n+1)}$ using $\mathcal{K}^{(n)}$, the current estimate of the scatter matrix \mathcal{K} :

$$\mathbf{x}^{(n+1)} \stackrel{\triangle}{=} \arg \min_{\mathbf{x} \geq \mathbf{0}} \text{KL}(\mathbf{d}, \mathcal{K}^{(n)} \mathcal{P}^c \mathbf{x} + \boldsymbol{\rho}). \quad (6.12)$$

- **Step 2** Get the current estimate of the scatter matrix $\mathcal{K}^{(n+1)}$ using $\mathbf{x}^{(n+1)}$, the current estimate of the emission mean vector \mathbf{x} :

$$\begin{aligned}\mathcal{K}^{(n+1)} &\stackrel{\triangle}{=} \arg \min_{\mathcal{K} \geq \mathbf{0}} \text{KL}(\mathbf{d}, \mathcal{K} \mathcal{P}^c \mathbf{x}^{(n+1)} + \boldsymbol{\rho}) \\ \text{subject to } &\sum_{i_1=1}^I \mathcal{K}_{i_1 i_2} \leq 1 \text{ for all } i_2.\end{aligned}\quad (6.13)$$

- **Step 3** Repeat the steps above until some chosen stopping criterion is met.

Note that the problem in **Step 1** can be solved using the MLEM algorithm [16, 26]. The reason is because Byrne showed that Shepp and Vardi's MLEM algorithm [16] minimizes the KL distance between the emission data \mathbf{d} and $\mathcal{P}^{\text{true}} \mathbf{x}$, where $\mathcal{P}^{\text{true}}$ is known. Specifically, given an initial guess $\mathbf{x}^{(n,0)} > \mathbf{0}$, the iteration for obtaining $\mathbf{x}^{(n+1)}$ is

$$x_j^{(n,m+1)} = \frac{x_j^{(n,m)}}{\sum_{i=1}^I [\mathcal{K}^{(n)} \mathcal{P}^c]_{ij}} \sum_{i=1}^I d_i \frac{[\mathcal{K}^{(n)} \mathcal{P}^c]_{ij} x_j^{(n,m)}}{[\mathcal{K}^{(n)} \mathcal{P}^c \mathbf{x}^{(n)} + \boldsymbol{\rho}]_i}, \quad j = 1, 2, \dots, J, \quad (6.14)$$

for $m = 0, 1, 2, \dots, M_1$. The current estimate of \mathbf{x} is defined to be $\mathbf{x}^{(n+1)} \stackrel{\triangle}{=} \mathbf{x}^{(n,M_1+1)}$, the M_1^{th} iterate of (6.14).

One of the issues surrounding the constrained minimization problem in (6.13) is that it is impossible to estimate the scatter matrix \mathcal{K} when all of the elements in the matrix are assumed to be unknown. The reason is because there would be too many unknowns, which would result in the problem being under-determined (note: the dimension of \mathcal{K} is $I \times I$, while there are only I data points). To reduce the number of unknowns in \mathcal{K} , we assume that $\mathcal{K}_{i_1 i_2} = 0$ if the detector pairs i_1 and i_2 are not in the same projection (see Figure 1–3 for the definition of a projection). This assumption means that an annihilation that would have been recorded by a detector pair within a certain projection if both photons had not undergone Compton scattering cannot be recorded by a detector pair within some other projection. Under the stated assumption, the number of unknown parameters in the matrix \mathcal{K} is dramatically reduced.

Moreover, \mathcal{K} is a block diagonal matrix with $T \times T$ sub-matrices along its diagonal

$$\mathcal{K} = \begin{bmatrix} \mathcal{K}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathcal{K}_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathcal{K}_S \end{bmatrix}, \quad (6.15)$$

where T is the number of detector pairs within each projection (e.g., 160), S is the number of projection angles (e.g., 192), and it is assumed that emission counts of the detector pairs within a projection are placed as a “chunk” in the emission data \mathbf{d} (i.e., $\mathbf{d} = [(1^{st} \text{ projection}), (2^{nd} \text{ projection}), \dots, (I^{th} \text{ projection})]'$).

Since \mathcal{K} is a block diagonal matrix, the minimization problem in (6.13) can be broken into S sub-problems. Consider the sinogram of \mathbf{d} , denoted by \mathcal{Y} , which is a $T \times S$ matrix. The s^{th} column of the sinogram \mathcal{Y} is the projection that corresponds to a certain projection angle. Figure 6-2 (a) shows an example of a sinogram. In the figure, the sinogram of the 14 minute emission data for plane 21 is shown. The $I \times 1$ vector $\mathcal{P}^c \mathbf{x}^{(n+1)}$ is known in the PET community as the forward projection of $\mathbf{x}^{(n+1)}$. We define a matrix $\mathcal{W}^{(n+1)}$ where the first column is the first S elements of $\mathcal{P}^c \mathbf{x}^{(n+1)}$ and the second column contains the $(S+1)^{th}$ to $(2S)^{th}$ elements of $\mathcal{P}^c \mathbf{x}^{(n+1)}$ and so on. In other words, the ordering of the detector pairs associated with $\mathcal{W}^{(n+1)}$ and \mathcal{Y} match. Figure 6-2 (b) shows an example of $\mathcal{W}^{(n+1)}$, where $\mathbf{x}^{(n+1)}$ was generated by running the MLEM algorithm for 1000 iterations on the emission data in Figure 6-2 (a). Under the assumption that $\mathcal{K}_{i_1 i_2} = 0$ if the detector pairs i_1 and i_2 are not in the same projection, the minimization problem in (6.13) can be expressed as: for $s = 1, 2, \dots, S$,

$$\begin{aligned} \mathcal{K}_s^{(n+1)} &\triangleq \arg \min_{\mathcal{K}_s \geq \mathbf{0}} \text{KL}(\mathbf{y}_s, \mathcal{K}_s \mathbf{w}_s^{(n+1)} + \mathbf{z}_s) \\ \text{subject to } &\sum_{i_1=1}^T [\mathcal{K}_s]_{i_1 i_2} \leq 1 \text{ for } i_2 = 1, 2, \dots, T, \end{aligned} \quad (6.16)$$

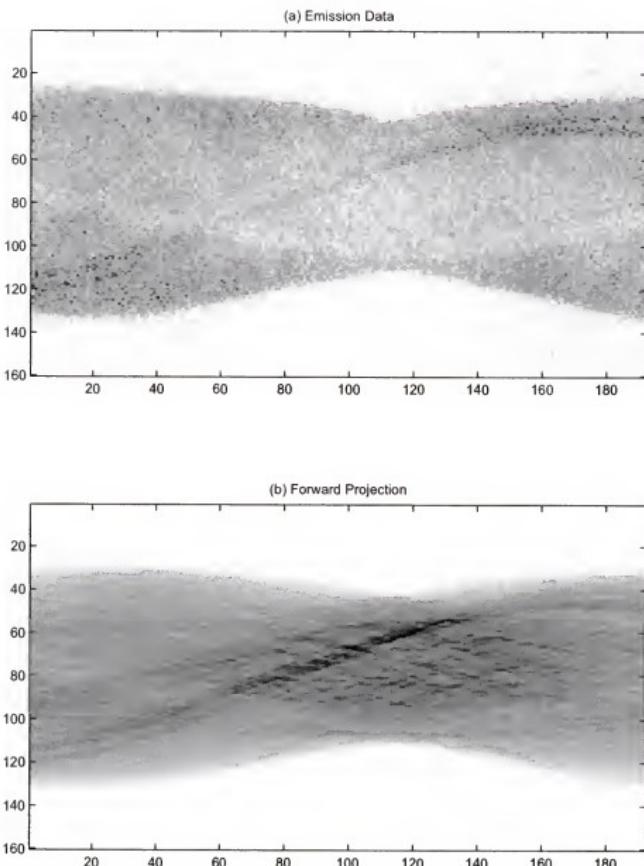


Figure 6.2: Sinograms: (a) 14 minute emission data for plane 21 (i.e., \mathcal{Y}) and (b) forward projection of $\mathbf{x}^{(n+1)}$ (i.e., $\mathcal{W}^{(n+1)} = \mathcal{P}^c \mathbf{x}^{(n+1)}$), where $\mathbf{x}^{(n+1)}$ is the 1000th MLEM iterate using the emission data in (a). Note that the images were adjusted with their own dynamic range.

where $\mathbf{z}_s \stackrel{\triangle}{=} [\rho_{\{(s-1)T+1\}} \ \rho_{\{(s-1)T+2\}} \ \dots \ \rho_{\{sT\}}]^T$, \mathcal{K}_s is the s^{th} sub-matrix of \mathcal{K} , and \mathbf{y}_s and $\mathbf{w}_s^{(n+1)}$ are the s^{th} column of \mathcal{Y} and $\mathcal{W}^{(n+1)}$, respectively.

Since there are $T \times T$ unknowns in each minimization problem in (6.16), there would be still too many unknowns. To reduce the number of unknowns, we assume that, for all s , \mathcal{K}_s is a Toeplitz matrix. An example of a Toeplitz matrix is

$$\begin{bmatrix} a_3 & a_2 & a_1 & 0 & 0 & 0 \\ a_4 & a_3 & a_2 & a_1 & 0 & 0 \\ a_5 & a_4 & a_3 & a_2 & a_1 & 0 \\ 0 & a_5 & a_4 & a_3 & a_2 & a_1 \\ 0 & 0 & a_5 & a_4 & a_3 & a_2 \\ 0 & 0 & 0 & a_5 & a_4 & a_3 \end{bmatrix}.$$

The assumption that \mathcal{K}_s is a Toeplitz matrix implies that there are at most T unknowns in each sub-matrix \mathcal{K}_s . Moreover, the assumption means that a single kernel can account for scatter within each projection.

The assumption that \mathcal{K}_s is a Toeplitz matrix can be justified for regions with approximately uniform attenuation, such as the brain. Consider Figure 6-3 in which a simplified PET scanner consisting of four detector pairs is depicted. In the figure, the dotted circle defines a region with uniform attenuation. We refer to the detector pairs (a_1, b_1) , (a_2, b_2) , (a_3, b_3) , and (a_4, b_4) as the first, second, third, and fourth detector pairs, respectively. Note that the geometry of the first and second detector pairs and the geometry of the second and third detector pairs are approximately same. Because of the approximately uniform attenuation of the subject and geometric similarity of the detector pairs, it can be said that the conditional probabilities \mathcal{K}_{21} and \mathcal{K}_{32} are approximately same. Using this rationale, for a projection of a PET scanner, it can be assumed that $\mathcal{K}_{i_2 i_1} \approx \mathcal{K}_{i_3 i_2}$, when $(i_2 - i_1) = (i_3 - i_2)$. Thus, we can construct \mathcal{K}_s that can be approximated as a Toeplitz matrix.

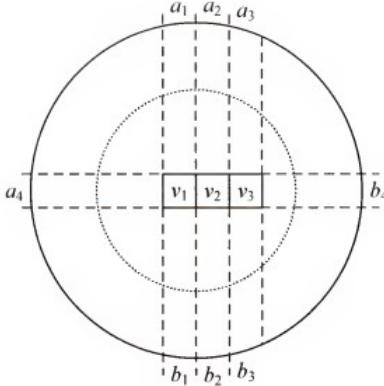


Figure 6.3: Geometry of a simplified PET image reconstruction problem: three voxels (v_1 , v_2 , and v_3) and four detector pairs ((a_1, b_1) , (a_2, b_2) , (a_3, b_3) , and (a_4, b_4)). The dashed lines define the tubes of the detector pairs. The dotted circle defines a region with uniform attenuation.

Now, consider the following functions: for an integer t ,

$$y_s(t) \triangleq \begin{cases} [y_s]_t, & t = 1, 2, \dots, T \\ 0, & \text{otherwise} \end{cases} \quad (6.17)$$

$$w_s^{(n+1)}(t) \triangleq \begin{cases} [w_s^{(n+1)}]_t, & t = 1, 2, \dots, T \\ 0, & \text{otherwise} \end{cases} \quad (6.18)$$

$$z_s(t) \triangleq \begin{cases} [z_s]_t, & t = 1, 2, \dots, T \\ 0, & \text{otherwise} \end{cases} \quad (6.19)$$

Under the assumption that \mathcal{K}_s is a Toeplitz matrix, $y_s(t)$ is approximately equal to the convolution of $w_s^{(n+1)}(t)$ and an unknown nonnegative function, denoted by $k_{(s,r)}(t)$, that depends on the s^{th} projection angle. Thus, for $t = 1, 2, \dots, T$,

$$y_s(t) \approx (k_{(s,r)} * w_s^{(n+1)})(t) + z_s(t) = \sum_{u=-\tau+1}^{\tau-1} k_{(s,r)}(u) w_s^{(n+1)}(t-u) + z_s(t), \quad (6.20)$$

where $k_{(s,\tau)}(t) \in [0, 1]$ for an integer t and $k_{(s,\tau)}(t) = 0$ for $|t| \geq \tau$. The parameter τ is defined by the user, but must satisfy the constraint $\tau \leq \lfloor (T+1)/2 \rfloor$. Observing (6.20), the product $k_{(s,\tau)}(u)w_s^{(n+1)}(t-u)$ equals the proportion of photon pairs that would have been recorded by detector pair $(t-u)$ if both photons produced by the annihilation had not undergone Compton scattering and they had propagated in exactly opposite directions, but instead are recorded by detector pair t . The parameter τ restricts the number of detector pairs to be convolved. In principle, the parameter τ is to be chosen in such a way that τ is proportional to the mean number of scatter events. Since the mean number of scatter events is proportional to the attenuation of the material, it is possible that attenuation map could be used to determine the parameter τ .

The convolution in (6.20) can be expressed in a matrix notation: for

$$(k_{(s,\tau)} * w_s^{(n+1)})(t) = [\mathcal{B}_s^{(n+1)} \mathbf{k}_{(s,\tau)}]_t, \quad t = 1, 2, \dots, T, \quad (6.21)$$

where $\mathbf{k}_{(s,\tau)} \stackrel{\triangle}{=} [k_{(s,\tau)}(-\tau+1) \ k_{(s,\tau)}(-\tau+2) \cdots k_{(s,\tau)}(\tau-1)]'$ and $\mathcal{B}_s^{(n+1)}$ is a known $T \times (2\tau - 1)$ matrix that is of the form

$$\mathcal{B}_s^{(n+1)} = \begin{bmatrix} [\mathbf{w}_s^{(n+1)}]_\tau & [\mathbf{w}_s^{(n+1)}]_{\tau-1} & \cdots & [\mathbf{w}_s^{(n+1)}]_1 & \cdots & 0 \\ [\mathbf{w}_s^{(n+1)}]_{\tau+1} & [\mathbf{w}_s^{(n+1)}]_\tau & \cdots & [\mathbf{w}_s^{(n+1)}]_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ [\mathbf{w}_s^{(n+1)}]_T & [\mathbf{w}_s^{(n+1)}]_{T-1} & \cdots & [\mathbf{w}_s^{(n+1)}]_{T-\tau+1} & \cdots & [\mathbf{w}_s^{(n+1)}]_{T-2\tau+2} \\ 0 & [\mathbf{w}_s^{(n+1)}]_T & \cdots & [\mathbf{w}_s^{(n+1)}]_{T-\tau+2} & \cdots & [\mathbf{w}_s^{(n+1)}]_{T-2\tau+3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & [\mathbf{w}_s^{(n+1)}]_T & \cdots & [\mathbf{w}_s^{(n+1)}]_{T-\tau+1} \end{bmatrix}.$$

Thus, \mathbf{y}_s can be approximated as

$$\mathbf{y}_s \approx \mathcal{B}_s^{(n+1)} \mathbf{k}_{(s,\tau)} + \mathbf{z}_s. \quad (6.22)$$

With (6.22), the minimization problem in (6.16) is now

$$\begin{aligned} \mathbf{k}_{(s,\tau)}^{(n+1)} &\stackrel{\Delta}{=} \arg \min_{\mathbf{k}_{(s,\tau)} \geq \mathbf{0}} \text{KL}(\mathbf{y}_s, \mathcal{B}_s^{(n+1)} \mathbf{k}_{(s,\tau)} + \mathbf{z}_s) \\ \text{subject to } & \sum_{t=1}^{2\tau-1} [\mathbf{k}_{(s,\tau)}]_t \leq 1 . \end{aligned} \quad (6.23)$$

The constrained optimization problem in (6.23) is difficult to solve because of the constraint $\sum_{t=1}^{2\tau-1} [\mathbf{k}_{(s,\tau)}]_t \leq 1$. Consequently, we first solve the following optimization problem:

$$\bar{\mathbf{k}}_{(s,\tau)}^{(n+1)} \stackrel{\Delta}{=} \arg \min_{\mathbf{k}_{(s,\tau)} \geq \mathbf{0}} \text{KL}(\mathbf{y}_s, \mathcal{B}_s^{(n+1)} \mathbf{k}_{(s,\tau)} + \mathbf{z}_s) . \quad (6.24)$$

Then, to get $\mathbf{k}_{(s,\tau)}^{(n+1)}$, we normalize $\bar{\mathbf{k}}_{(s,\tau)}^{(n+1)}$ so that the sum of its elements equals one:

$$[\mathbf{k}_{(s,\tau)}^{(n+1)}]_t \stackrel{\Delta}{=} \frac{[\bar{\mathbf{k}}_{(s,\tau)}^{(n+1)}]_t}{\sum_{t=1}^{2\tau-1} [\bar{\mathbf{k}}_{(s,\tau)}^{(n+1)}]_t} \quad t = 1, 2, \dots, (2\tau - 1) . \quad (6.25)$$

The minimization problem in (6.24) can be solved by the MLEM algorithm [16, 26] because it has the same form as the optimization problem in **Step 1**. Specifically, given an initial estimate $\bar{\mathbf{k}}_{(s,\tau)}^{(n,0)} > \mathbf{0}$, the iteration is as follows: for $m = 0, 1, \dots, M_2$,

$$[\bar{\mathbf{k}}_{(s,\tau)}^{(n,m+1)}]_j = \frac{[\bar{\mathbf{k}}_{(s,\tau)}^{(n,m)}]_j}{\sum_{i=1}^T [\mathcal{B}_s^{(n+1)}]_{ij}} \sum_{i=1}^T \frac{[\mathcal{B}_s^{(n+1)}]_{ij} [\mathbf{y}_s]_i}{[\mathcal{B}_s^{(n+1)}]_{ij} \bar{\mathbf{k}}_{(s,\tau)}^{(n,m)} + \mathbf{z}_s]_i} , \quad j = 1, 2, \dots, (2\tau - 1) . \quad (6.26)$$

We define $\bar{\mathbf{k}}_{(s,\tau)}^{(n+1)} \stackrel{\Delta}{=} \bar{\mathbf{k}}_{(s,\tau)}^{(n,M_2+1)}$ and normalize $\bar{\mathbf{k}}_{(s,\tau)}^{(n+1)}$ using (6.25) to get $\mathbf{k}_{(s,\tau)}^{(n+1)}$.

Given $\mathbf{k}_{(s,\tau)}^{(n+1)}$, we determine $\mathcal{K}_s^{(n+1)}$ by using the following equation:

$$\mathcal{K}_s^{(n+1)} = \begin{bmatrix} [\mathbf{k}_{(s,\tau)}^{(n+1)}]_\tau & [\mathbf{k}_{(s,\tau)}^{(n+1)}]_{\tau-1} & \cdots & [\mathbf{k}_{(s,\tau)}^{(n+1)}]_1 & \cdots & 0 \\ [\mathbf{k}_{(s,\tau)}^{(n+1)}]_{\tau+1} & [\mathbf{k}_{(s,\tau)}^{(n+1)}]_\tau & \cdots & [\mathbf{k}_{(s,\tau)}^{(n+1)}]_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ [\mathbf{k}_{(s,\tau)}^{(n+1)}]_{2\tau-1} & [\mathbf{k}_{(s,\tau)}^{(n+1)}]_{2\tau-2} & \cdots & [\mathbf{k}_{(s,\tau)}^{(n+1)}]_\tau & \cdots & 0 \\ 0 & [\mathbf{k}_{(s,\tau)}^{(n+1)}]_{2\tau-1} & \cdots & [\mathbf{k}_{(s,\tau)}^{(n+1)}]_{\tau+1} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & [\mathbf{k}_{(s,\tau)}^{(n+1)}]_\tau \end{bmatrix}. \quad (6.27)$$

Finally, $\mathcal{K}^{(n+1)}$ is defined as a block diagonal matrix with $\{\mathcal{K}_s^{(n+1)}\}$ along its diagonal:

$$\mathcal{K}^{(n+1)} = \begin{bmatrix} \mathcal{K}_1^{(n+1)} & 0 & 0 & 0 \\ 0 & \mathcal{K}_2^{(n+1)} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \mathcal{K}_S^{(n+1)} \end{bmatrix}. \quad (6.28)$$

Repeating Steps 1 and 2 generate the estimates of \mathcal{K} and \mathbf{x} . Once the estimate of \mathcal{K} is available, denoted by $\hat{\mathcal{K}}$, a regularized image reconstruction algorithm, such as the PML, APML, and QEP algorithm, can be used to estimate the emission mean vector \mathbf{x} . More specifically, we first let our estimate for $\mathcal{P}^{\text{true}}$ be $\hat{\mathcal{P}}^{\text{true}} = \hat{\mathcal{K}}\mathcal{P}^c$. Then, the probability matrix $\hat{\mathcal{P}}^{\text{true}}$ is used in the image reconstruction algorithm of choice.

In summary, the Steps of the PCiPS algorithm are: for $n = 0, 1, 2, \dots$

- **Step 1** Get an initial estimate for the emission mean vector $\mathbf{x}^{(0)} > \mathbf{0}$ and an initial estimate for the scatter matrix $\mathcal{K}^{(0)}$
- **Step 2** Get the current estimate of the emission mean vector $\mathbf{x}^{(n+1)}$ using $\mathcal{K}^{(n)}$, which is the current estimate of the scatter matrix \mathcal{K} :

$$\mathbf{x}^{(n+1)} = \arg \min_{\mathbf{x} \geq \mathbf{0}} \text{KL}(\mathbf{d}, \mathcal{K}^{(n)} \mathcal{P}^c \mathbf{x} + \rho). \quad (6.29)$$

- **Step 3** For $s = 1, 2, \dots, S$, get $\bar{\mathbf{k}}_{(s,\tau)}^{(n+1)}$ using (6.26).

- **Step 4** For $s = 1, 2, \dots, S$, normalize $\bar{\mathbf{k}}_{(s,\tau)}^{(n+1)}$ using (6.25) to get $\mathbf{k}_{(s,\tau)}^{(n+1)}$.
- **Step 5** For $s = 1, 2, \dots, S$, get $\mathcal{K}_s^{(n+1)}$ using (6.27).
- **Step 6** Get $\mathcal{K}^{(n+1)}$ using (6.28).
- **Step 7** Repeat **Steps 2** through **6** for a chosen number, M , of iterations.
- **Step 8** Define $\hat{\mathcal{P}}^{\text{true}} \triangleq \mathcal{K}^{(M+1)}\mathcal{P}^c$ and use $\hat{\mathcal{P}}^{\text{true}}$ in the APML algorithm or some other algorithm of choice.

CHAPTER 7

SIMULATIONS AND EXPERIMENTAL STUDY

To evaluate the algorithms in Chapters 3, 4, and 5, and the method in Chapter 6, we applied them to real thorax phantom data and compared them quantitatively and qualitatively to certain existing algorithms. Also, in Section 7.1 simulation studies with computer-generated synthetic data are presented for the PML and QEP algorithms in Chapters 3 and 5, respectively. It should be noted that simulation results with the synthetic data have limitations because they are generated under the assumption that the system model for emission data in Section 1.4 is exactly correct. However, the system model is not perfect due to errors discussed in Section 1.3.

Thorax phantom data was obtained from the PET laboratory at the Emory University School of Medicine. The phantom was filled with 2-[¹⁸F]fluoro-2-deoxy-d-glucose ([¹⁸F]FDG) and scanned using a Siemens-CTI ECAT EXACT [model 921] scanner in slice-collimated mode (i.e., septa-extended mode). Thirty independent data sets were generated from multiple scans of duration 7 minutes. Fifteen realizations of 14 min data were generated by adding non-overlapping two 7 min data sets. The [¹⁸F]FDG concentration for the heart wall, heart cavity, liver, three tumors, and thorax cavity of the thorax phantom by Data Spectrum Inc., were $0.72\mu\text{Ci}/\text{ml}$, $0.23\mu\text{Ci}/\text{ml}$, $0.72\mu\text{Ci}/\text{ml}$, $2.01\mu\text{Ci}/\text{ml}$, and $0.24\mu\text{Ci}/\text{ml}$, respectively. The lungs, which contained styrofoam beads, were filled with a $0.25\mu\text{Ci}/\text{ml}$ solution of [¹⁸F]FDG. The concentrations were chosen to mimic those observed in whole-body scans. The tumors were of size 1 cm, 1.5 cm, and 2 cm. The sinogram consists of 160 radial bins and 192 angles. The physical dimensions of the image space is $43.9 \times 43.9 \text{ cm}^2$ and the reconstructed images contain 128×128 voxels (voxel size is 3.43×3.43

mm^2). Two planes (10 and 21) were considered in the experiments. Plane 10 contains activity due to the heart, lungs, spine, and background, while plane 21 contains activity due to the heart, two tumors (1.5 cm and 2.0 cm), and background. The total number of prompts for planes 10 and 21 was 397,000 and 340,000, respectively, for 14 minute data. The randoms makeup about 10% and 12% of the data for planes 10 and 21, respectively.

The probability matrix \mathcal{P} was computed using the angle-of-view method [16] with corrections for errors due to attenuation and detector inefficiency. To get the attenuation correction factors, post-injected transmission scan data was collected for three minutes and the attenuation correction method by Anderson *et al.* [9] was employed. A normalization file was used to correct for detector inefficiency. Finally, the randoms were used as noise free estimates of the mean numbers of accidental coincidences.

For all of the experiments and simulations, we used a uniform initial estimate (all voxels equal $\sum_i d_i/J$), the eight nearest neighbors of the j^{th} voxel were used for N_j , and the weights, $\{w_{jk}\}$, are one for horizontal and vertical nearest neighbors and $1/\sqrt{2}$ for diagonal nearest neighbors.

In Section 7.1, we applied the PML and QEP algorithms to real thorax phantom data and computer-generated synthetic data, and compared them quantitatively and qualitatively. Then, performance of the APML algorithm will be evaluated in Section 7.2. Finally, experimental results with the probability estimation method in Chapter 6 will be presented in Section 7.3.

7.1 Regularized Image Reconstruction Algorithms

In this section, we compare the PML and QEP algorithms, quantitatively and qualitatively, to the MLEM algorithm and a penalized weighted least-squares (PWLS) algorithm [42]. Two ad-hoc forms of regularization include the post-filtering of MLEM estimates and early termination of the MLEM algorithm (usually quite far from the

MLEM estimate). Given their simplicity, we also compared the post-filtering (MLEM-F) and early-stopping (MLEM-S) strategies to the proposed algorithms.

Quantitative comparisons were made using contrast as a figure-of-merit:

$$\text{Contrast} \triangleq \frac{|M_{ROI} - M_B|}{M_B}, \quad (7.1)$$

where M_{ROI} and M_B denote the mean of a chosen region-of-interest (ROI) and background, respectively. We define another figure-of-merit that quantifies the distinguishability of the two tumors:

$$\text{Distinguishability} \triangleq \frac{|M_T - M_I|}{|M_T - M_B|}, \quad (7.2)$$

where M_T and M_I denote the mean activity of the two tumor regions and intermediate region between the two tumors, respectively. If two tumors overlap each other, then $M_I = M_T$ and the distinguishability will be zero. On the other hand, if intermediate region between the two tumors has the same mean as the background, then the distinguishability will be one.

For image comparisons, converged images were used for the MLEM, PML, and PWLS algorithms. The QEP images result by running the QEP algorithm for the same number of iterations as the PML algorithm. This was necessary because the QEP algorithm does not have a single objective function. Recall that the QEP algorithm defines a new objective function to be minimized at each iteration. For the PML, QEP, and PWLS images, the penalty parameter, β , was chosen in such a way that the standard deviation of their soft-tissue (i.e., background) regions were equal. In this sense, the algorithms are “balanced” with respect to β . For the MLEM-S and MLEM-F images, early MLEM iterates and filtered converged MLEM images were obtained that matched the standard deviation of the soft-tissue regions in the PML, QEP, and PWLS images. To filter the MLEM image, we used 5×5 Gaussian filters with different standard deviation values.

7.1.1 Synthetic Data

In this subsection, we present simulation results for software phantoms. Fifty realizations were used in the simulation study. To generate emission data, a software phantom was forward projected (i.e., $\mathcal{P}\mathbf{x}$) using the \mathcal{P} matrix, where it was assumed that there are no errors except accidental coincidences. Then, for each bin, the prompts and randoms were generated using pseudo-random Poisson variates with mean $[\mathcal{P}\mathbf{x}]_i + \rho$ and ρ , respectively. The constant ρ was chosen such that the mean accidental coincidence rate was approximately 10%. The mean number of prompts and randoms are about 550,000 and 50,000, respectively. For simulation studies, dimensions of the image space follow the corresponding ones of the real phantom image space. The total number of intensities within a software phantom was about 500,000.

We first consider a tumor phantom. Figure 7.1 shows a software phantom that consists of two tumors (1.7 cm and 2.4 cm in diameter) in a uniform circular background with a diameter of 30.5 cm, where two tumors and background intensities are 7×74 and 74, respectively (tumor contrast is 6). The image also depicts regions for contrast and distinguishability calculation. The intermediate region between the two tumors consists of 14 voxels and the large and small tumors consist of 45 and 21 voxels, respectively.

For the tumor phantom in Figure 7.1, we used the log-cosh penalty function $\lambda(t) = \log(\cosh(\frac{t}{\delta}))$ with $\delta = 20$ in the PML and QEP algorithms, while the quadratic penalty function was used in the PWLS algorithm. For the QEP algorithm, $\eta(t) = \xi \tanh(\frac{t}{\xi})$ was used with $\xi = 150$ unless noted. The parameters δ and ξ were chosen experimentally.

Figure 7.2 is a plot of the images obtained by the MLEM, MLEM-S, MLEM-F, PML, QEP, and PWLS algorithms when the tumor phantom in Figure 7.1 was used.

The MLEM image is the 500th MLEM iterate and 200 iterations were used to reconstruct the PML, QEP, and PWLS images. For the PML, QEP, and PWLS images, β was chosen such that the standard deviation of the background was approximately 12. The MLEM-S image is the 20th MLEM iterate. To get the MLEM-F image, the MLEM image in Figure 7 2 (a) was filtered once using a 5×5 Gaussian filter with standard deviation of 1.27 in voxels. As stated, all of the images in Figure 7 2 have the same background standard deviation except for the MLEM image. The MLEM image in Figure 7 2 (a) is considerably noisy (background standard deviation is about 78) compared to the other images. Figures 7 2 (d) and (e) illustrate that the PML image and the QEP image are smooth and, at the same time, the tumors in the images are resolvable and differ greatly from the background. On the other hand, Figures 7 2 (c) and (f) demonstrate that the images generated by the MLEM-F and PWLS algorithm are too smooth, especially near the boundary of the tumors. Figure 7 2 (b) shows that edges of the tumors in the MLEM-S image are not as clear as the ones in the PML and QEP images. In Figure 7 3, the images in Figure 7 2 are plotted with their own dynamic range.

For the images in Figure 7 2, the contrast of the QEP image was -1%, 12%, 21%, 4%, and 22% higher than the MLEM, MLEM-S, MLEM-F, PML, and PWLS images, respectively, for the large tumor. The increased contrast of the QEP image for the small tumor with respect to the MLEM, MLEM-S, MLEM-F, PML, and PWLS images was -4%, 16%, 31%, 5%, and 34%, respectively. The increased tumor distinguishability of the QEP image with respect to the MLEM, MLEM-S, MLEM-F, PML, and PWLS images was -5%, 16%, 34%, 5%, and 40%, respectively. The MLEM image outperformed the QEP image in contrast and distinguishability comparison. However, as can be seen in Figure 7 2 (a), the MLEM image is extremely noisy.

Figures 7 4 (a) and (b) are line plots (the row is shown in Figure 7 1) of the images in Figure 7 2. For the row under consideration, it can be seen from the line plots that the PML and QEP images have a higher degree of contrast than the other images, except for the MLEM image. And, the edges in the QEP image are sharper than those in the PML image. As expected, the MLEM image is excessively noisy from the line plot.

Figures 7 5 (a) and (b) are plots of the average contrast of the large tumor and small tumor versus the average background standard deviation using fifty realizations, respectively, when the tumor phantom in Figure 7 1 was used. Further, a plot of the average standard deviation of the large tumor versus the average background standard deviation for the fifty realizations is shown in Figure 7 5 (c). Finally, in Figure 7 5 (d), a plot of the average distinguishability of two tumors versus the average background standard deviation for fifty realizations is shown. As can be seen from Figures 7 5 (a), (b), and (d), the contrast curves and the distinguishability curve of the QEP algorithm lie above the curves of the other algorithms for comparable “background noise”. Thus, for a fixed level of comparable background noise, the QEP images, on average, have the greatest contrast and distinguishability. The average standard deviation curves of the PML and QEP algorithms in Figure 7 5 (c) generally lie below the corresponding curves of the other algorithms for reasonably small background noise.

To see where the PML and QEP algorithms break down in terms of contrast, we performed simulations using the synthetic tumor phantom in Figure 7 1 with four different tumor contrast values (tumor contrast equals 3, 1.5, 0.75, and 0.5). For the PML and QEP algorithms, $\beta = 1/16$ and $1/32$ were used, respectively, and 200 iterations were used. For the QEP algorithm, $\xi = 150$ was used for tumor contrast of 3, whereas $\xi = 80$ was used for the other tumor contrast values (i.e., tumor contrast equals 1.5, 0.75, and 0.5). The parameters β and ξ were chosen

experimentally. Figures 7 6 (a) and (b) are the PML and QEP images, respectively, obtained by using the phantom in Figure 7 1 with tumor contrast of 3. As can be seen in the figures, when tumor contrast was 3, the tumors in the PML and QEP images are clearly resolvable. Figures 7 6 (c) and (d) are the PML and QEP images, respectively, obtained by using the phantom in Figure 7 1 with tumor contrast of 1.5. From the figures, the tumors in the PML and QEP images are clear enough when tumor contrast was 1.5. It should be mentioned that the images in Figure 7 6 have the same standard deviation of the background that is approximately 10. Figures 7 7 (a) and (b) are the PML and QEP images, respectively, obtained by using the phantom in Figure 7 1 with tumor contrast of 0.75. From the figures, the tumors in the PML and QEP images are still resolvable when tumor contrast was 0.75. Consider the PML and QEP images in Figures 7 7 (c) and (d), respectively, that are obtained by using the phantom in Figure 7 1 with tumor contrast of 0.5. The tumors in the PML and QEP images are hardly resolvable which implies that the PML and QEP algorithms break down when tumor contrast was 0.5. The images in Figure 7 7 have the same standard deviation of the background that is approximately 9.

Figures 7 8 (a), (b), (c), and (d) are plots of the average contrast of the small tumor versus the average background standard deviation using fifty realizations, when tumor contrast of the phantom in Figure 7 1 were 3, 1.5, 0.75, and 0.5, respectively. As can be seen from Figures 7 8 (a) and (b), the contrast curves of the QEP algorithm lie above the curves of the PML algorithm for a fixed background noise when tumor contrast equals 3 and 1.5. For tumor contrast of 0.75 and 0.5, the curves of the PML and QEP algorithm coincide as can be seen in Figures 7 8 (c) and (d) which implies that the PML and QEP algorithms perform similar to each other when tumor contrast is small enough.

To see where the PML and QEP algorithms break down in terms of spacing between two tumors, we performed simulations using four different synthetic tumor

phantoms where each phantom has different tumor spacing. We compared the PML and QEP images both visually and in terms of distinguishability. Figures 7 9 (a), (b), and (c) show software phantoms that consist of two tumors (each tumor equals 3×3 voxels). For the phantoms in Figures 7 9 (a), (b), and (c), tumor contrast is 3 and the spacing between two tumors is 2, 3, and 4 voxels, respectively. Figure 7 9 (d) shows a software phantoms that consists of two tumors (each tumor equals 3×3 voxels) with the spacing of 2 voxels, where tumor contrast is 6. The images in Figure 7 9 also depict regions for distinguishability calculation. The intermediate regions between the two tumors consist of 6, 8, 10, and 6 voxels for the phantoms in Figures 7 9 (a), (b), (c) and (d), respectively.

Figure 7 10 is a plot of the PML and QEP images obtained by using the phantoms in Figures 7 9 (a) and (b). Figure 7 11 is a plot of the PML and QEP images obtained by using the phantoms in Figures 7 9 (c) and (d). The PML and QEP images in Figures 7 10 and 7 11 were from 200 iterations. The PML and QEP images in Figures 7 10 and 7 11 have the same standard deviation of the background that is approximately 10 and 9, respectively. For the PML and QEP images in Figures 7 10 and 7 11, $\beta = 1/16$ and $1/32$ were used, respectively. Figure 7 10 indicates that the PML and QEP algorithms generate images where the tumors are clearly separated when the spacing between the tumors is 3 and 4 in voxels. As can be seen in Figures 7 11 (a) and (b), the PML and QEP algorithms were not able to resolve the two tumors when the spacing between the tumors is 2 in voxels. However, when the tumor contrast was increased, the PML and QEP algorithms worked well when the spacing between the tumors is 2 in voxels as shown in Figures 7 11 (c) and (d).

Figure 7 12 is a plot of the average distinguishability of two tumors versus the average background standard deviation using fifty realizations, when the tumor phantoms in Figure 7 9 were used. As can be seen from Figures 7 12 (a), (b), (c), and (d), the distinguishability curves of the QEP algorithm lie above the curves of the



Figure 7 1: A software tumor phantom is shown. Contrast of the tumors in the phantom is 6. The regions surrounded by the dotted and dashed lines define the tumor intermediate region (i.e., M_I) and background region, respectively.

PML algorithm for a fixed background noise. Thus, for a fixed level of background noise, the QEP images, on average, have greater distinguishability.

7.1.2 Real Data

In this subsection, we present experimental results using real phantom data for plane 21. Unless noted, the data are from 14 minute scans. The image in Figure 7 13, which was produced by averaging fifteen converged MLEM images, depicts the regions that were used in the contrast and distinguishability calculations. In the large and small tumor ROIs and intermediate region, the number of voxels were 24, 12, and 16, respectively.

In the PML and QEP algorithms, we used the log-cosh penalty function $\lambda(t) = \log(\cosh(\frac{t}{\delta}))$ with $\delta = 50$. Since noise in real data is stronger than the synthetic data due to errors (e.g., scatter), we increased the value of the parameter δ to reduce background noise (observe that we used $\delta = 20$ for the synthetic phantom in Figure 7 1). For the QEP algorithm, $\eta(t) = \xi \tanh(\frac{t}{\xi})$ was used with $\xi = 500$ and 1000 for 7 minute and 14 minute real data, respectively. We varied ξ because the data sets lead to reconstructed images that have different edge “heights”. As in the simulations

(a) MLEM

(b) MLEM-S

(c) MLEM-F

(d) PML

(e) QEP

(f) PWLS

Figure 7-2: Comparison of emission images when the synthetic phantom in Figure 7-1 was used: (a) MLEM image, (b) MLEM-S image, (c) MLEM-F image, (d) PML image, (e) QEP image, and (f) PWLS image. The images in (a) and (b) are from 500 and 20 iterations, respectively, while the images in (d), (e), and (f) are from 200 iterations. The image in (c) was obtained from filtering the MLEM image once with a 5×5 Gaussian filter with standard deviation of 1.27 in voxels. For the PML, QEP, and PWLS images, β was chosen in such a way that the standard deviation of the background is approximately 12. Specifically, $\beta = 0.0415$, 0.021, and 0.006 for the PML, QEP, and PWLS images, respectively. The standard deviation of the background of images in (b) and (c) is also approximately 12. For display purposes, all the images were adjusted so that they have the same dynamic range.

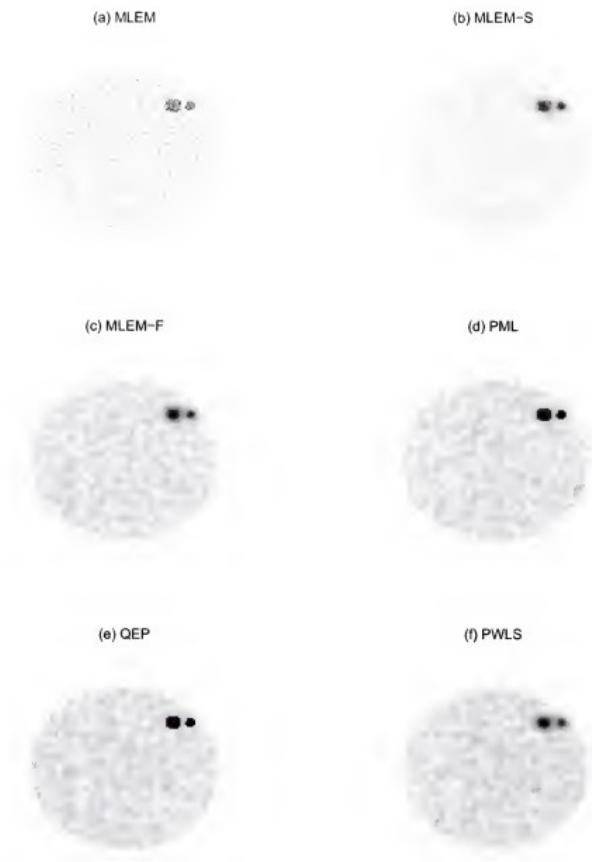


Figure 7-3: The images in Figure 7-2 are shown with their own dynamic range.

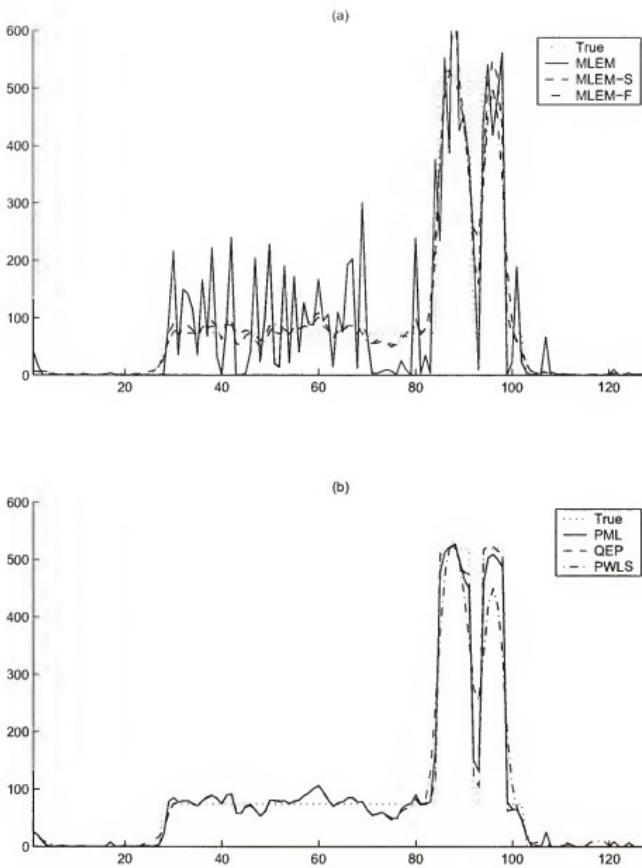


Figure 7-4: A line plot comparison of the reconstructed images in Figures 7-2 (a), (b), and (c) is shown in (a). A line plot comparison of the reconstructed images in Figures 7-2 (d), (e), and (f) is shown in (b).

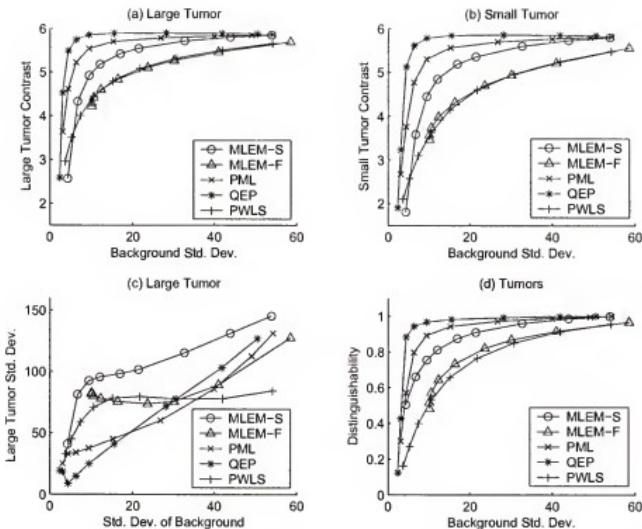


Figure 7-5: Plots of the average contrast of the large and small tumors versus the average background standard deviation are shown in (a) and (b), respectively, when the synthetic phantom in Figure 7-1 was used. A plot of the average standard deviation of the large tumor versus the average background standard deviation is shown in (c). In (d), a plot of the average distinguishability of two tumors versus the average background standard deviation is shown. Fifty synthetic data realizations were used in the study. For the MLEM-S curves, the images from iterations 5 – 160 were used. For the MLEM-F curves, the MLEM images were filtered once by 5×5 Gaussian filters with a standard deviation range of $0.44 - 3.0$ voxels (each voxel is $3.43 \times 3.43 \text{ mm}^2$). For the PML, QEP, and PWLS algorithms, the images were reconstructed using two hundred iterations for $\beta = 2^{-1} - 2^{-9}$, $2^{-1} - 2^{-9}$, and $2^{-4} - 2^{-12}$, respectively.

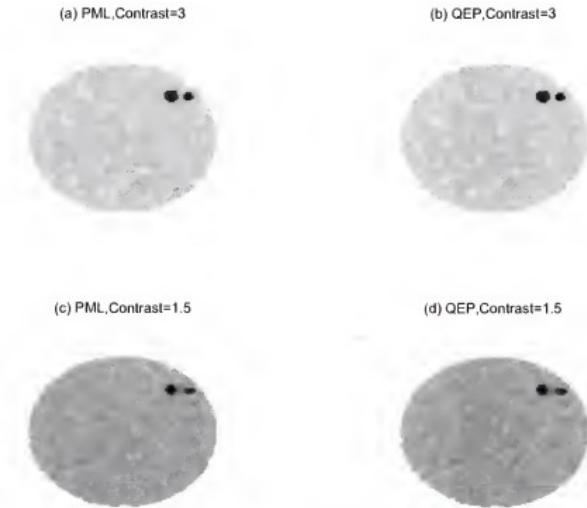


Figure 7-6: Comparison of emission images: (a) PML image and (b) QEP image when tumor contrast of the phantom in Figure 7-1 was 3, and (c) PML image and (d) QEP image when tumor contrast of the phantom in Figure 7-1 was 1.5. All images are from 200 iterations and they have the same background standard deviation that is approximately 10. For the PML and QEP images, $\beta = 1/16$ and $1/32$ were used, respectively. For the QEP images in (b) and (d), $\xi = 150$ and $\xi = 80$ were used, respectively. For display purposes, each set of images (i.e., (a) and (b), (c) and (d)) were adjusted so that they have the same dynamic range.

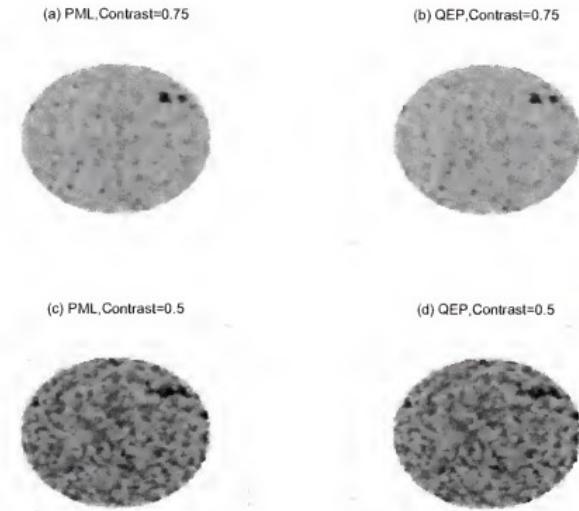


Figure 7-7: Comparison of emission images: (a) PML image and (b) QEP image when tumor contrast of the phantom in Figure 7-1 was 0.75, and (c) PML image and (d) QEP image when tumor contrast of the phantom in Figure 7-1 was 0.5. All images are from 200 iterations and they have the same background standard deviation that is approximately 10. For the PML and QEP images, $\beta = 1/16$ and $1/32$ were used, respectively. For the QEP images in (b) and (d), $\xi = 80$ was used. For display purposes, each set of images (i.e., (a) and (b), (c) and (d)) were adjusted so that they have the same dynamic range.

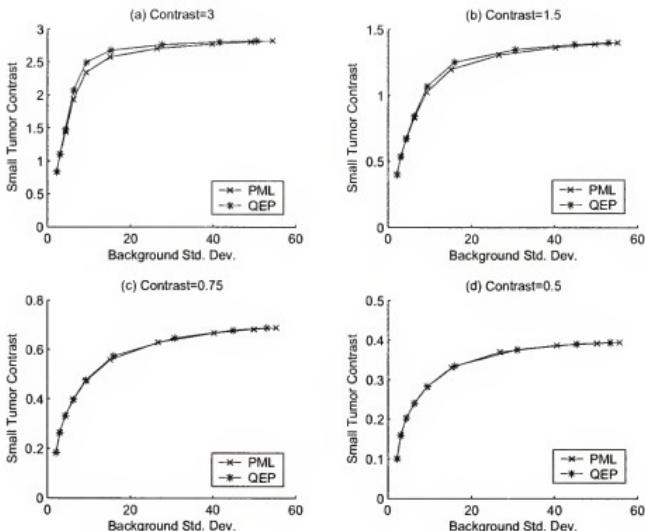


Figure 7-8: Plots of the average contrast of the small tumor versus the average background standard deviation are shown in (a), (b), (c), and (d) when the synthetic phantom in Figure 7-1 was used with tumor contrast of 3, 1.5, 0.75 and 0.5, respectively. Fifty synthetic data realizations were used in the study. For the PML and QEP algorithms, the images were reconstructed using two hundred iterations for $\beta = 2^{-1} - 2^{-9}$. For the QEP curves in (b), (c), and (d), $\xi = 80$ was used, whereas $\xi = 150$ was used for the QEP curves in (a).

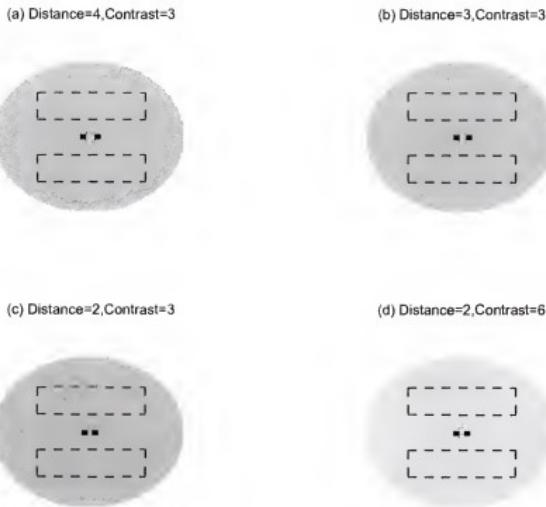


Figure 7-9: Software tumor phantoms are shown for different spacing between two tumors. For the phantoms in (a), (b), (c), and (d), spacing between tumors are 2, 3, 4, and 2 in voxels, respectively. Tumor contrast is 3, 3, 3, and 6 for the phantoms in (a), (b), (c), and (d), respectively. The regions surrounded by the dotted and dashed lines define the tumor intermediate region (i.e., M_I) and background region, respectively. The images are shown with their own dynamic range.

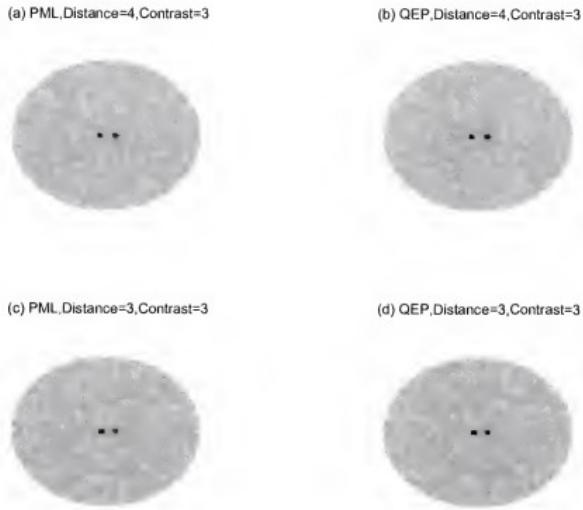


Figure 7-10: Comparison of emission images: (a) PML image and (b) QEP image when the phantom in Figure 7-9 (a) was used, and (c) PML image and (d) QEP image when the phantom in Figure 7-9 (b) was used. All images are from 200 iterations and they have the same standard deviation of the background that is approximately 10. For the PML and QEP images, $\beta = 1/16$ and $1/32$ were used, respectively. For display purposes, each set of images (i.e., (a) and (b), (c) and (d)) were adjusted so that they have the same dynamic range.

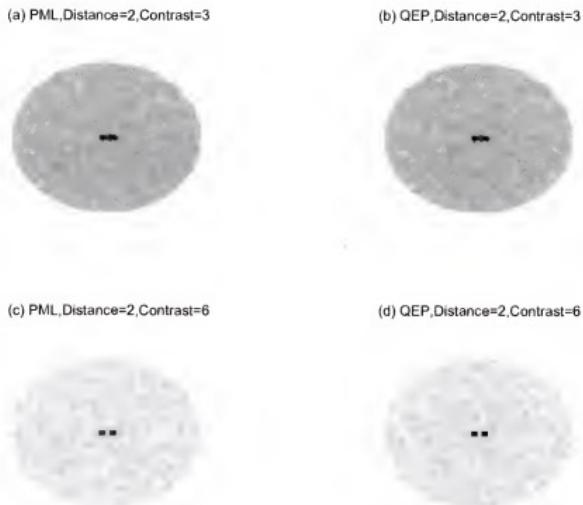


Figure 7-11: Comparison of emission images: (a) PML image and (b) QEP image when the phantom in Figure 7-9 (c) was used, and (c) PML image and (d) QEP image when the phantom in Figure 7-9 (d) was used. All images are from 200 iterations and they have the same standard deviation of the background that is approximately 9. For the PML and QEP images, $\beta = 1/16$ and $1/32$ were used, respectively. For display purposes, each set of images (i.e., (a) and (b), (c) and (d)) were adjusted so that they have the same dynamic range.

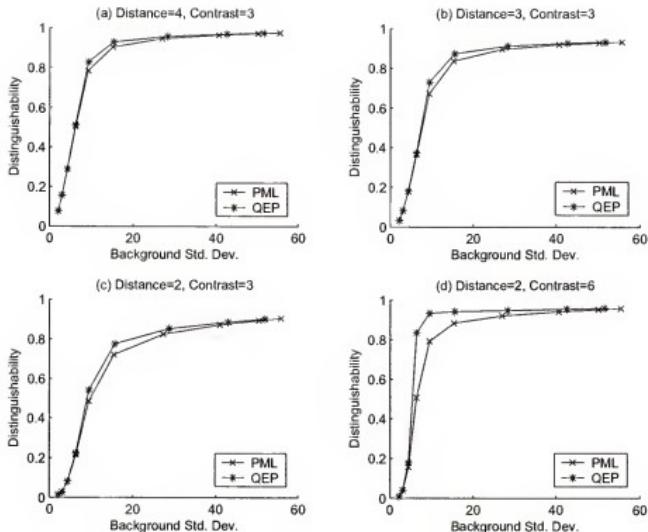


Figure 7-12: Plots of the average distinguishability of two tumors versus the average background standard deviation are shown in (a), (b), (c), and (d) when the synthetic phantoms in Figures 7-9 (a), (b), (c), and (d) were used, respectively. Fifty synthetic data realizations were used in the study. For the PML and QEP algorithms, the images were reconstructed using two hundred iterations for $\beta = 2^{-1} - 2^{-9}$.

with the synthetic phantom, the quadratic penalty function was used in the PWLS algorithm. Note, the parameters δ and ξ were chosen experimentally.

Figure 7 14 is a plot of the images for plane 21 that were obtained by the MLEM, MLEM-S, MLEM-F, PML, QEP, and PWLS algorithms. The number of iterations used to reconstruct the MLEM, PML, and PWLS images was 500, 200, and 200, respectively. Figure 7 15 shows that the PML and QEP images for different iteration numbers when a 14 minute data set for plane 21 was used. As can be seen in Figure 7 15, early iterates of the PML and QEP algorithms can be used because 100^{th} iterates are not much different from 500^{th} iterates. As in the simulated data, the number of iterations for the QEP algorithm was set to the number of iterations used for the PML algorithm.

For the PML, QEP, and PWLS images in Figure 7 14, β was chosen so that the standard deviation of their backgrounds are approximately same (background standard deviation is approximately 68). Using 9 iterations and repeating the application of a 5×5 Gaussian filter two times yielded the MLEM-S and MLEM-F images with a background standard deviation of 68. Figures 7 14 (d) and (e) illustrate that the tumors in the PML image and the QEP image are resolvable and differ significantly from the background. On the other hand, the tumors are not as distinct in the images produced by the MLEM-S, MLEM-F, and PWLS algorithms (see Figures 7 14 (b), (c), and (f)). As expected, the MLEM image in Figure 7 14 (a) is considerably noisier and grainier than the other images. In Figure 7 16, the images in Figure 7 14 are plotted with their own dynamic range.

The true contrast of the small and large tumors was 7.38. Due to finite resolution effects, all of the algorithms underestimate the contrast [68]. Consequently, the algorithm that produces the greatest contrast is to be preferred. For the images in Figure 7 14, the large tumor contrast of the QEP image was -14% , 37% , 32% , 3% , and 34% higher than the MLEM, MLEM-S, MLEM-F, PML, and PWLS images,

respectively. The increased contrast of the QEP image for the small tumor with respect to the MLEM, MLEM-S, MLEM-F, PML, and PWLS images was -28% , 46% , 46% , 9% , and 43% , respectively. The increased tumor distinguishability of the QEP image with respect to the MLEM, MLEM-S, MLEM-F, PML, and PWLS images was -8% , 20% , 49% , 5% , and 35% , respectively. Although the contrast for “true” MLEM images (i.e., images obtained when the MLEM algorithm is not terminated early) can be quite large, the noise level of these images limits their practical use.

Figure 7 17 is a line plot (the row is shown in Figure 7 13) of the images in Figures 7 14 (b), (d), (e), and (f) (the image in Figure 7 14 (a) is too noisy and the image in Figure 7 14 (c) is too smooth). For the row under consideration, it can be seen from the line plots that the PML and QEP images have a higher degree of contrast than the other images. And, the edges in the QEP image are sharper than those in the PML image.

Figures 7 18 (a) and (b) are plots of the average contrast of the large tumor and small tumor versus the average background standard deviation using fifteen realizations for plane 21, respectively. Further, plots of the average standard deviation of the large tumor and the average distinguishability of two tumors versus the average background standard deviation for the fifteen realizations are shown in Figures 7-18 (c) and (d), respectively. Just like the contrast curves in the synthetic data simulations, the contrast curves of the QEP algorithm lie above the corresponding curves of the other algorithms. Thus, for a fixed level of background noise, the QEP images, on average, have the greatest contrast. Not too surprising, the large tumor average standard deviation curves of the QEP algorithm in Figure 7-18 (c) generally lie above the corresponding curves of the PML algorithm. The PWLS algorithm produced large tumor standard deviation curves that lie below the corresponding curves generated by the other algorithms. However, the degree of smoothing produced by the PWLS algorithm was too great. This claim is supported qualitatively by Figure

7 14 (f) and quantitatively by the contrast plots in Figures 7 18 (a) and (b). The distinguishability curves of the PML and QEP algorithms lie above the curves of the other algorithms when the background standard deviation is less than 300. Above that point, the distinguishability curves of the PML and QEP algorithms lie below the curves of the other algorithms except the PWLS curve. However, images with background standard deviation greater than 300 are too noisy for practical use. Thus, for practical noise levels, the PML and QEP images, on average, have the greatest contrast and distinguishability.

To see how the algorithms would perform in shorter duration protocols, we now consider fifteen realizations of 7 minute real phantom data for plane 21. Figures 7 19 (a) and (b) are plots of the average contrast of the large tumor and small tumor versus the average background standard deviation using fifteen realizations, respectively. A plot of the average standard deviation of the large tumor versus the average background standard deviation for the fifteen realizations is shown in Figure 7 19 (c). Also, in Figure 7 19 (d), we provide a plot of the average distinguishability of two tumors versus the average background standard deviation for fifteen realizations. As in the experiments with the 14 minute real phantom data, it is evident that the PML and QEP algorithms outperform the MLEM-S, MLEM-F, and PWLS algorithms in terms of contrast recovery and tumor distinguishability.

Figure 7 20 is a plot of the images obtained by the MLEM, MLEM-S, MLEM-F, PML, QEP, and PWLS algorithms for a 7 minute data set for plane 21. The number of iterations used to reconstruct the MLEM, PML, QEP, and PWLS images was 500, 200, 200, and 200, respectively. For the PML, QEP, and PWLS images in Figure 7 20, β was chosen so that the standard deviation of their backgrounds are approximately 38. Using 8 iterations and repeating the application of a 5×5 Gaussian filter two times yielded the MLEM-S and MLEM-F images with a background standard deviation of 38. As in the experiments with the 14 minute data, the tumors in the PML image



Figure 7-13: The mean of 15 MLEM emission images reconstructed from 14 minute data for plane 21. The boxes indicate the regions used to compute the contrast and distinguishability of the tumors. The solid and dashed lines define the tumors and background regions, respectively. The region surrounded by the dotted lines defines the tumor intermediate region (i.e., M_I). The dotted line indicates the row chosen for the line plots.

and the QEP image are resolvable and differ significantly from the background. In Figure 7-21, the images in Figure 7-20 are plotted with their own dynamic range.

For the images in Figure 7-20, the large tumor contrast of the QEP image was -20% , 37% , 25% , 5% , and 25% higher than the MLEM, MLEM-S, MLEM-F, PML, and PWLS images, respectively. The increased contrast of the QEP image for the small tumor with respect to the MLEM, MLEM-S, MLEM-F, PML, and PWLS images was -58% , 34% , 27% , 5% , and 23% , respectively. The increased tumor distinguishability of the QEP image with respect to the MLEM, MLEM-S, MLEM-F, PML, and PWLS images was -30% , 12% , 42% , 6% , and 25% , respectively. Figure 7-22 is a line plot (the row is shown in Figure 7-13) of the images in Figures 7-20 (b), (d), (e), and (f). As in the experiments with the 14 minute data, it can be seen from the line plots that the PML and QEP images have a higher degree of contrast than the other images.

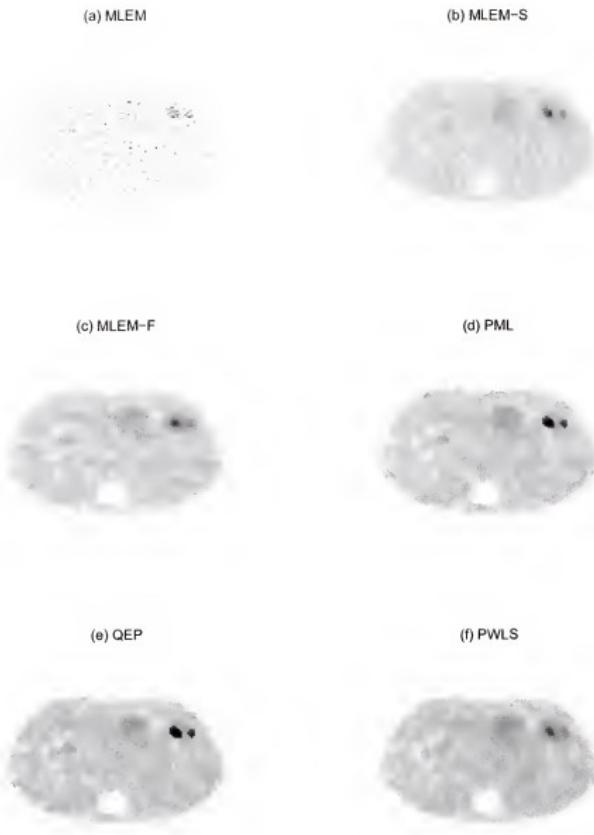


Figure 7-14: Comparison of emission images when a 14 minute real phantom data for plane 21 was used: (a) MLEM image, (b) MLEM-S image, (c) MLEM-F image, (d) PML image, (e) QEP image, and (f) PWLS image. The images in (a) and (b) are from 500 and 9 iterations, respectively, while the images in (d), (e), and (f) are from 200 iterations. The image in (c) was obtained from filtering the MLEM image two times with a 5×5 Gaussian filter with standard deviation of 1.95 in voxels. For the PML, QEP, and PWLS images, β was chosen in such a way that the standard deviation of the background is approximately 68. Specifically, $\beta = 2^{-6}$, 2^{-7} , and 0.00017 for the PML, QEP, and PWLS images, respectively. Note, the standard deviation of the background of images in (b) and (c) is also approximately 68. For display purposes, all the images were adjusted so that they have the same dynamic range except the MLEM image because it has very wide dynamic range.

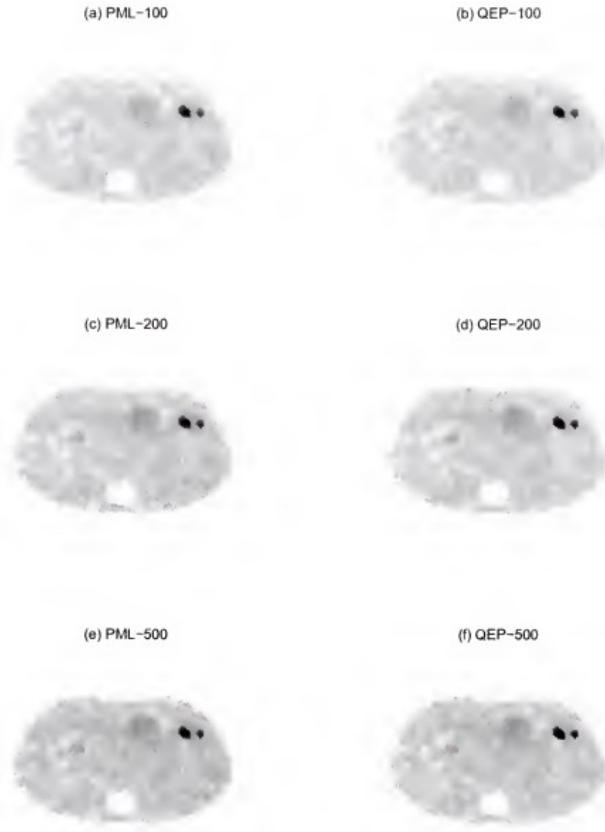


Figure 7-15: Iteration comparison of emission images reconstructed from using a 14 minute real phantom data for plane 21: (a) PML image using 100 iterations, (b) QEP image using 100 iterations, (c) PML image using 200 iterations, (d) QEP image using 200 iterations, (e) PML image using 500 iterations, and (f) QEP image using 500 iterations. For the PML and QEP images, $\beta = 2^{-6}$ and 2^{-7} , respectively. For display purposes, all the images were adjusted so that they have the same dynamic range.

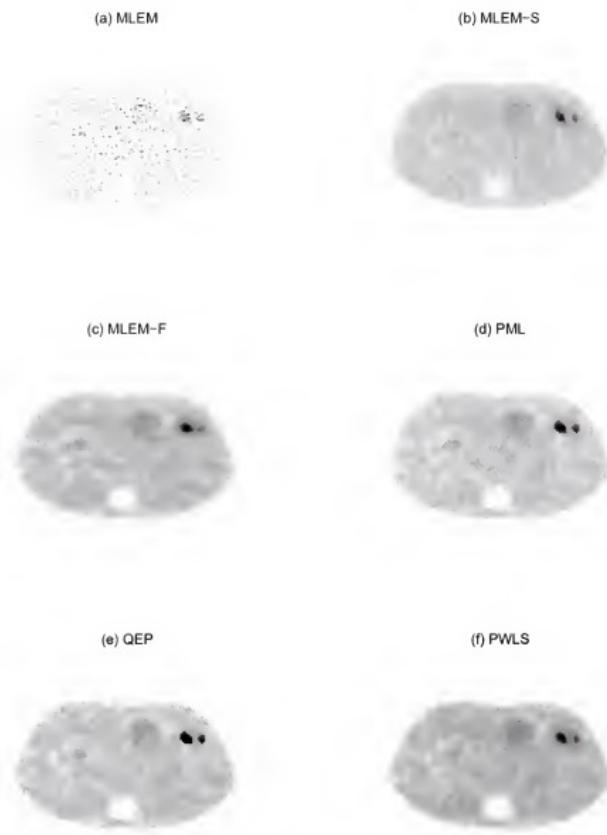


Figure 7-16: Emission images in Figure 7-14 are shown with their own dynamic range.

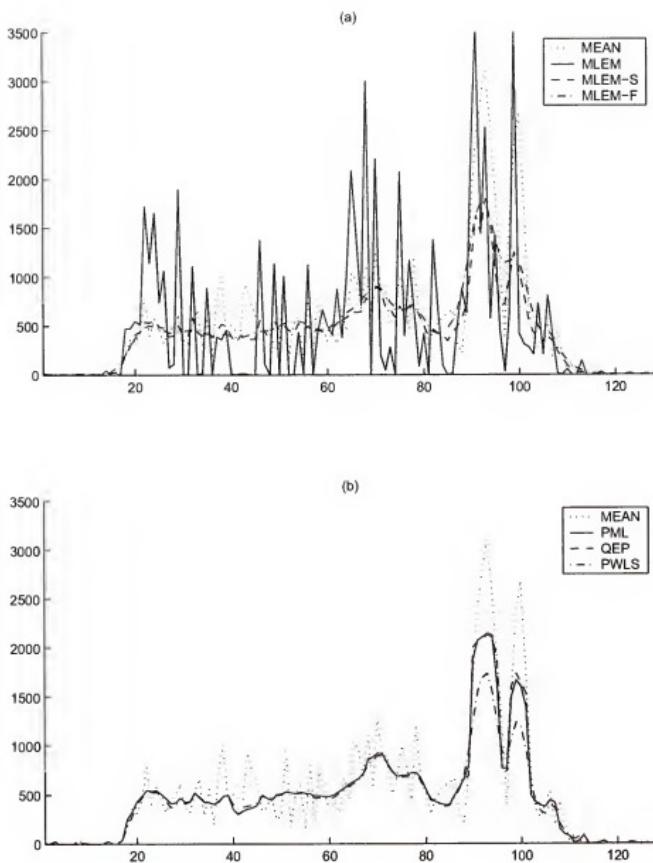


Figure 7-17: A line plot comparison of the reconstructed images in Figures 7-14 (a), (b), and (c) is shown in (a). A line plot comparison of the reconstructed images in Figures 7-14 (d), (e), and (f) is shown in (b).

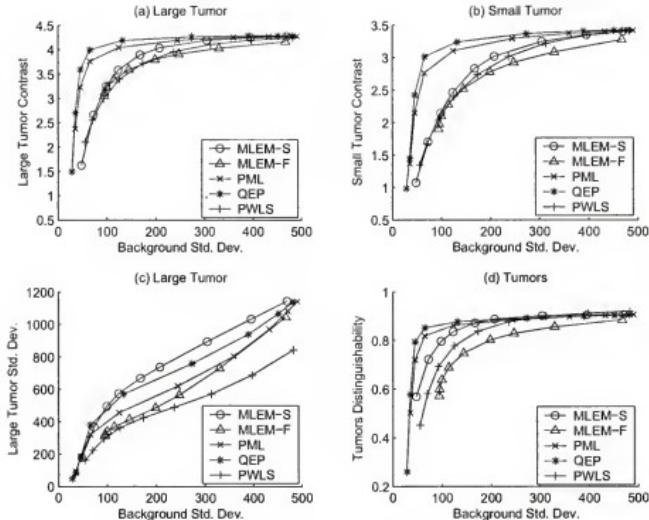


Figure 7-18: Plots of the average contrast of the large and small tumors versus the average background standard deviation are shown in (a) and (b), respectively. A plot of the average standard deviation of the large tumor versus the average background standard deviation is shown in (c). In (d), a plot of the average distinguishability of two tumors versus the average background standard deviation is shown. Fifteen realizations were used and 14 minute real phantom data for plane 21 was used in the study. For the MLEM-S curves, the images from iterations 5 – 160 were used. For the MLEM-F curves, the MLEM images were filtered once by 5×5 Gaussian filters with a standard deviation range of 0.44 – 3.0 voxels (each voxel is $3.43 \times 3.43 \text{ mm}^2$). For the PML, QEP, and PWLS algorithms, the images were reconstructed using two hundred iterations for $\beta = 2^{-4} - 2^{-12}$, $2^{-4} - 2^{-12}$, and $2^{-12} - 2^{-20}$, respectively.

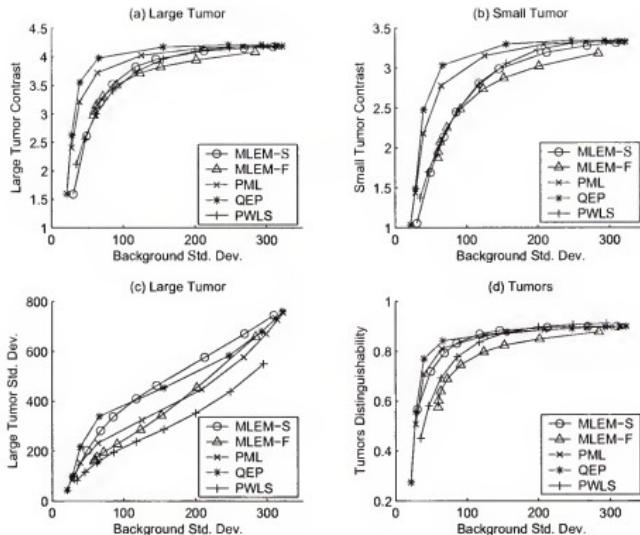


Figure 7.19: Plots of the average contrast of the large and small tumors versus the average background standard deviation are shown in (a) and (b), respectively. A plot of the average standard deviation of the large tumor versus the average background standard deviation is shown in (c). In (d), a plot of the average distinguishability of two tumors versus the average background standard deviation is shown. Fifteen realizations were used and 7 minute real phantom data for plane 21 was used in the study. For the MLEM-S curves, the images from iterations 5 – 160 were used. For the MLEM-F curves, the MLEM images were filtered once by 5×5 Gaussian filters with a standard deviation range of $0.44 - 3.0$ voxels (each voxel is $3.43 \times 3.43 \text{ mm}^2$). For the PML, QEP, and PWLS algorithms, the images were reconstructed using two hundred iterations for $\beta = 2^{-4} - 2^{-12}$, $2^{-4} - 2^{-12}$, and $2^{-12} - 2^{-20}$, respectively.

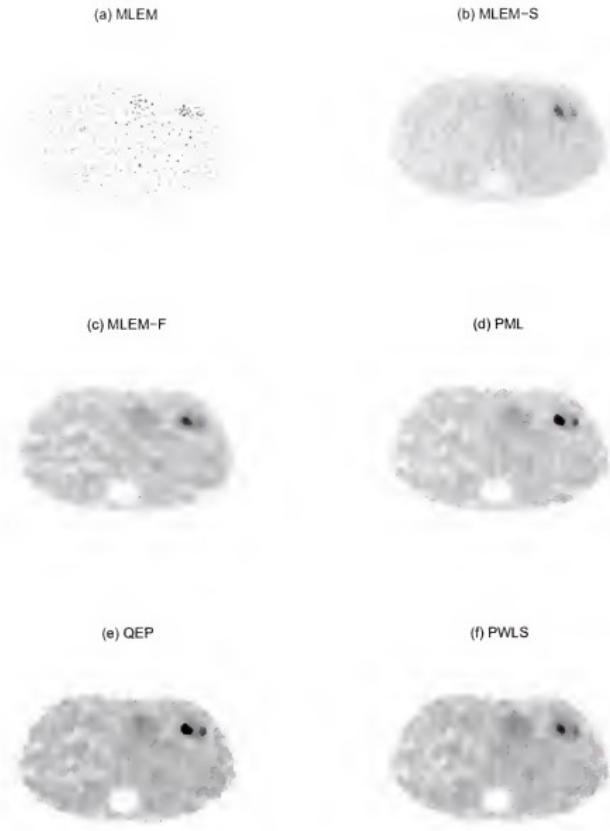


Figure 7-20: Comparison of emission images when a 7 minute real phantom data for plane 21 was used: (a) MLEM image, (b) MLEM-S image, (c) MLEM-F image, (d) PML image, (e) QEP image, and (f) PWLS image. The images in (a) and (b) are from 500 and 8 iterations, respectively, while the images in (d), (e), and (f) are from 200 iterations. The image in (c) was obtained from filtering the MLEM image two times with a 5×5 Gaussian filter with standard deviation of 1.95 in voxels. For the PML, QEP, and PWLS images, β was chosen in such a way that the standard deviation of the background is approximately 38. Specifically, $\beta = 0.029$, 0.0145 , and 0.000155 for the PML, QEP, and PWLS images, respectively. Note, the standard deviation of the background of images in (b) and (c) is also approximately 38. For display purposes, all the images were adjusted so that they have the same dynamic range except the MLEM image because it has very wide dynamic range.

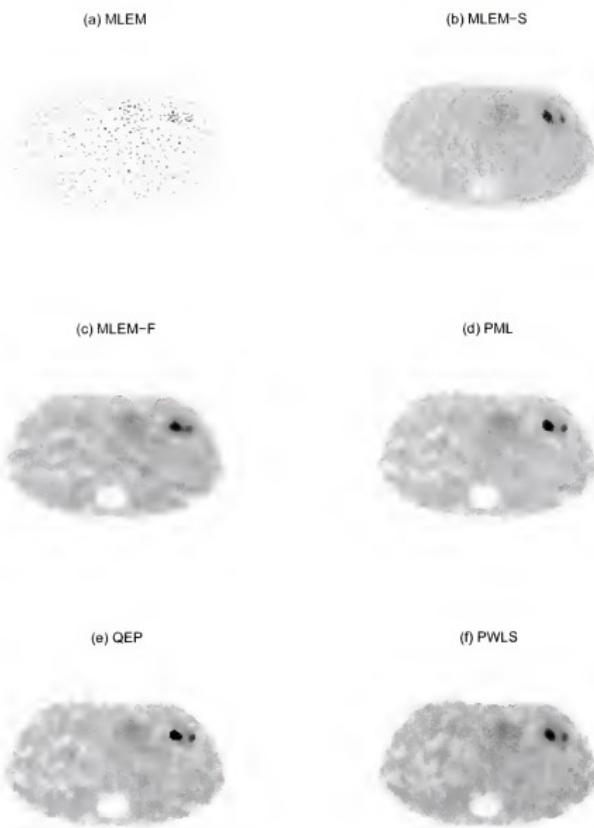


Figure 7.21: Emission images in Figure 7.20 are shown with their own dynamic range.

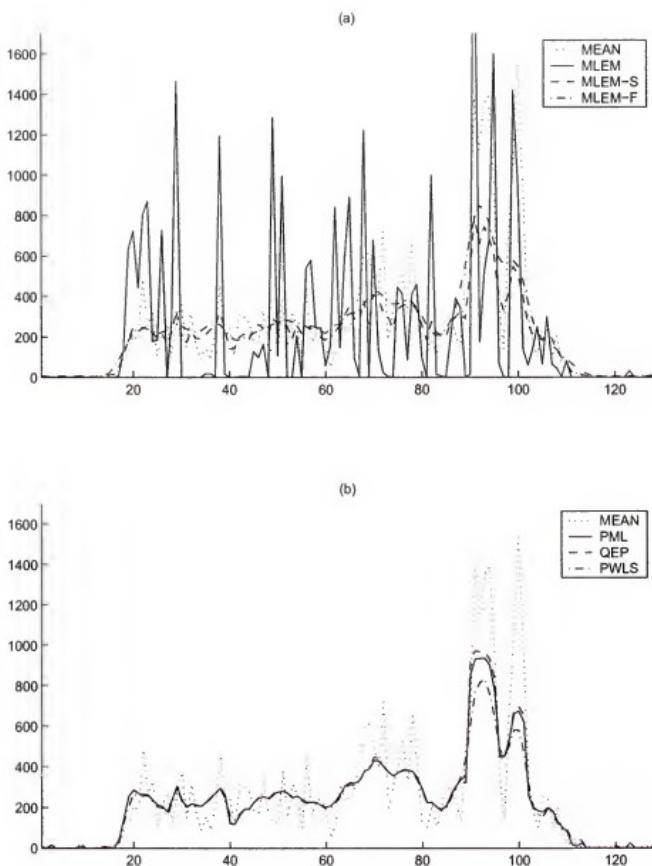


Figure 7-22: A line plot comparison of the reconstructed images in Figures 7-20 (a), (b), and (c) is shown in (a). A line plot comparison of the reconstructed images in Figures 7-20 (d), (e), and (f) is shown in (b).

7.2 APML algorithm

To evaluate the APML algorithm in Chapter 4, we applied it to plane 21 from the real thorax phantom data (unless noted, the data are from 14 minute scans) and compared it to the PML algorithm and an algorithm by Ahn and Fessler [35] called the block sequential regularized expectation-maximization (BSREM)-II algorithm. The BSREM-II algorithm is a straightforward modification of Ahn and Fessler's BSREM-I algorithm [35]. The BSREM-II algorithm results by setting any negative element of a BSREM-I iterate to a small positive number. The BSREM-I algorithm is based on the BSREM algorithm by De Pierro and Yamagishi [33] and the ordered-subsets idea originally put forth by Hudson and Larkin [45].

We used the penalty parameter $\beta = 0.02$ and 0.04 for 14 minute data and 7 minute data, respectively, and the log-cosh function $\lambda(t) = \log(\cosh(\frac{t}{\delta}))$ with $\delta = 50$ as the penalty function. The values of β and δ were chosen experimentally such that the reconstructed images were visually "good". For the BSREM-II algorithm, 8 and 24 subsets, and the ordering rule suggested by Ahn and Fessler [35] (i.e., make projections in two successive subsets as perpendicular as possible each other) were used. The ordering rule was originally introduced by Herman and Meyer [69]. Additionally, the relaxation parameter rule specified by Ahn and Fessler [35] was used in the implementation.

In Figure 7-23, plots of the cost versus CPU-time are shown for the APML algorithm for different values of ϵ when 14 minute data were used. As can be seen in the figure, the convergence rate of the APML algorithm depends on ϵ . Moreover, $\epsilon = 0$ generated the slowest convergence rate for the considered planes. These claims are supported by Figure 7-24 in which plots of the number of iterations versus ϵ are shown for the APML algorithm to decrease the PML objective function to $\Phi(\mathbf{x}^*)$, where \mathbf{x}^* is the 5000th iterate of the PML algorithm. Practically speaking, for the planes considered, \mathbf{x}^* is the minimizer of the PML objective function because the PML

algorithm did not decrease the PML objective function after about 1000 iterations until 5000 iterations. For $0 < \epsilon \leq 0.01$, the number of iterations required for the APML algorithm to decrease the PML objective function to $\Phi(\mathbf{x}^*)$ varied from 83 to 125 and 113 to 185 for planes 10 and 21, respectively. For $\epsilon = 0$, the number of iterations required were 359 and 652 for planes 10 and 21, respectively. For $0 < \epsilon \leq 0.01$, the decreased convergence time (in CPU seconds) of the APML algorithm with respect to the PML algorithm was 71% – 81% and 75% – 85% for planes 10 and 21, respectively. For $\epsilon = 0$, the decreased convergence time of the APML algorithm with respect to the PML algorithm was 17% and 12% for planes 10 and 21, respectively.

In Figure 7.25, plots of the cost versus CPU-time are shown for the PML, APML, and BSREM-II algorithms. As can be seen in the figure, the APML algorithm for $\epsilon = 0.01$ decreased the PML objective function more than about three times faster than the PML algorithm. The APML algorithm for $\epsilon = 0.01$ needed about 27% and 18% of the CPU-time that was necessary for the PML algorithm to decrease the PML objective function to $\Phi(\mathbf{x}^*)$ for planes 10 and 21, respectively. At the early iterations (CPU-time less than about 10 seconds), the BSREM-II algorithm for 8 subsets produced the greatest decrease. However, after about 10 seconds in CPU-time, the APML algorithm for $\epsilon = 0.01$ decreased the PML objective function faster than the other algorithms. It should be pointed out that the BSREM-II algorithm did not decrease the PML objective function to $\Phi(\mathbf{x}^*)$ until 10,000 iterations, whereas $\Phi(\mathbf{x}^*)$ was obtained by the APML algorithm for $\epsilon = 0.01$ with 116 and 136 iterations for planes 10 and 21, respectively. Moreover, the convergence rate of the BSREM-II algorithm significantly depends on the number of subsets as shown in Figure 7.25.

To decrease the PML objective function faster at the early iterations, an alternative would be to first use the ordered subsets algorithm for a few iterations and then switch to the APML algorithm. To do this, we divide the emission data into a few subsets (e.g., 8 or 24) according to the ordering rule by Herman and Meyer [69].

For the r^{th} subset, we define a sub-objective function as in [35]:

$$\Phi_r(\mathbf{x}) \triangleq \sum_{i \in \mathbb{M}_r} \{[\mathcal{P}\mathbf{x}]_i - d_i \log([\mathcal{P}\mathbf{x} + \boldsymbol{\rho}]_i) + \rho_i + \log(d_i!)\} + \frac{\beta}{R} \Lambda(\mathbf{x}), \quad (7.3)$$

where R is the number of subsets, \mathbb{M}_r is the set of indices corresponding to emission data within the r^{th} subset, and $r = 1, 2, \dots, R$. Note that $\Phi(\mathbf{x}) = \sum_{r=1}^R \Phi_r(\mathbf{x})$. For each subset and the corresponding sub-objective function, a sub-iteration is performed sequentially. At the $(n, r)^{th}$ sub-iteration, a surrogate function for the r^{th} sub-objective function is constructed using (3.29), (3.30), (3.31), and (3.32) at the $(n, r)^{th}$ sub-iterate $\mathbf{x}^{(n,r)}$. The next sub-iterate $\mathbf{x}^{(n,r+1)}$ is defined to be the nonnegative minimizer of the surrogate function. After one pass of the entire sub-iterations, we define $\mathbf{x}^{n+1} \triangleq \mathbf{x}^{(n,R+1)}$ and $\mathbf{x}^{(n+1,1)} \triangleq \mathbf{x}^{n+1}$ for the next pass of sub-iterations. We refer to the above iteration as the ordered subset PML (OS-PML) iteration.

Figure 7.26 shows plots of the cost versus CPU-time for the BSREM-II algorithm for 8 subsets and the APML algorithm with a few OS-PML iterations (2, 3, and 4) for 8 and 24 subsets. As can be seen in the figure, the APML algorithm with a few OS-PML iterations decreases the PML objective function faster than the BSREM-II algorithm for the early iterations. Specifically, the APML algorithm with the ordered-subsets idea needed less than about 8 seconds in CPU-time to decrease the PML objective function to the cost that the BSREM-II algorithm decreased with 10 seconds. The convergence rate depends on the number of OS-PML iterations and the number of subsets, but the degree of dependency is not much great as shown in the figure.

In Figure 7.27, iterates for plane 10 produced by the BSREM-II algorithms for 8 subsets and the APML algorithm with 2 OS-PML iterations for 24 subsets are shown for different CPU-times. As shown in the figure, the iterates of the APML and BSREM-II algorithms look very similar each other even though their associated costs are different. Figure 7.28 shows iterates for plane 21 produced by the BSREM-II

algorithms for 8 subsets and the APML algorithm with 2 OS-PML iterations for 24 subsets are shown for different CPU-times.

Now, we consider 7 minute real phantom data. In Figure 7.29, plots of the cost versus CPU-time are shown for the PML, APML, and BSREM-II algorithms when 7 minute data were used. As can be seen in the figure, the convergence rates of the algorithms are similar to the ones for 14 minute data except the BSREM-II algorithm with 24 subsets that converged much slower than the other algorithms. Again, this indicates that the convergence rate of the BSREM-II algorithm significantly depends on the number of subsets. Figure 7.30 shows plots of the cost versus CPU-time for the BSREM-II algorithm for 8 subsets and the APML algorithm with a few OS-PML iterations (2, 3, and 4) for 8 and 24 subsets when 7 minute data were used. As in the experiments with the 14 minute data, the APML algorithm with a few OS-PML iterations decreases the PML objective function faster than the BSREM-II algorithm with 8 subsets for the early iterations. For 7 minute data case, we do not include the corresponding images because the visual comparisons were similar to those observed in the images for 14 minute data.

To see whether convergence rates in Figures 7.25, 7.26, 7.29, and 7.30 are “typical”, we averaged convergence rate for fifteen realization. Since cost varies for each realization, we used the normalized-cost-difference that is defined as

$$\frac{\Phi(\tilde{\mathbf{x}}^{(n)}) - \Phi(\tilde{\mathbf{x}}^{(100)})}{\Phi(\tilde{\mathbf{x}}^{(0)}) - \Phi(\tilde{\mathbf{x}}^{(100)})}, \quad (7.4)$$

where $\tilde{\mathbf{x}}^{(0)}$ and $\tilde{\mathbf{x}}^{(100)}$ are the uniform initial estimate and the 100th APML iterate, respectively. The reason why we used $\tilde{\mathbf{x}}^{(100)}$ is that the APML algorithm “almost” converges with 100 iterations, which means that the PML objective function does not decrease appreciably after 100 APML iterations. In Figures 7.31 and 7.32, plots of the average normalized-cost-difference (averaged normalized-cost-difference over fifteen realizations) versus CPU-time are shown for the PML, APML, and BSREM-II

algorithms when fifteen 7 and 14 minute data were used, respectively. Figures 7.33 and 7.34 show plots of the average normalized-cost-difference versus CPU-time for the BSREM-II algorithm for 8 subsets and the APML algorithm with a few OS-PML iterations (2, 3, and 4) for 8 and 24 subsets when fifteen 7 and 14 minute data were used, respectively. The figures confirm that the observed convergence rates in Figures 7.25, 7.26, 7.29, and 7.30 are “typical”.

Table 7.1 contains the CPU-times, memory accesses, floating point operations, and the number of function calls (\sqrt{t} and $\lambda(t)$) per iteration for the algorithms used in the experiments. From the table, it is shown that the APML algorithm greatly reduced the number of iterations for convergence compared with the PML algorithm. Convergence in the table means that an algorithm decreases the PML objective function until it equals the 5000th iterate of the PML algorithm, denoted by \mathbf{x}^* . Although the BSREM-II algorithm did not converge until 10,000 iterations, the algorithm “almost” converged at the point because $|\Phi(\tilde{\mathbf{x}}^{(1000)}) - \Phi(\mathbf{x}^*)| \approx 1$, where $\tilde{\mathbf{x}}^{10000}$ is the 10000th BSREM-II iterate (note: $|\Phi(\tilde{\mathbf{x}}^{(1000)}) - \Phi(\tilde{\mathbf{x}}^{(10000)})| \approx 1$).

7.3 PCiPS Algorithm

To evaluate the probability matrix estimation method in Chapter 6, we applied the probability correction in projection space (PCiPS) algorithm to plane 21 from the real thorax phantom data (unless noted, the data are from 14 minute scans) and compared it to the APML algorithm with a geometric probability matrix \mathcal{P}^c . The probability matrix \mathcal{P}^c was computed using the angle-of-view method [16] with corrections for errors due to attenuation and detector inefficiency. As in Sections 7.1 and 7.2, to get the attenuation correction factors, post-injected transmission scan data was collected for three minutes and the attenuation correction method by Anderson *et al.* [9] was employed. A normalization file was used to correct for detector inefficiency. Finally, the randoms were used as noise free estimates of the mean numbers of accidental coincidences. The PCiPS algorithm used one iteration to estimate the

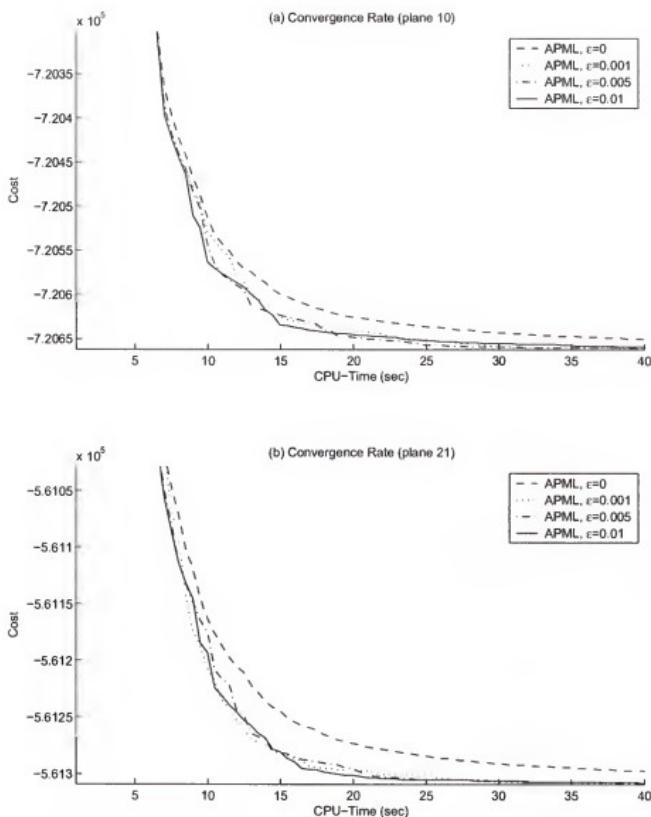


Figure 7-23: Convergence rate comparison of the APML algorithm for different values of ϵ when 14 minute data were used.

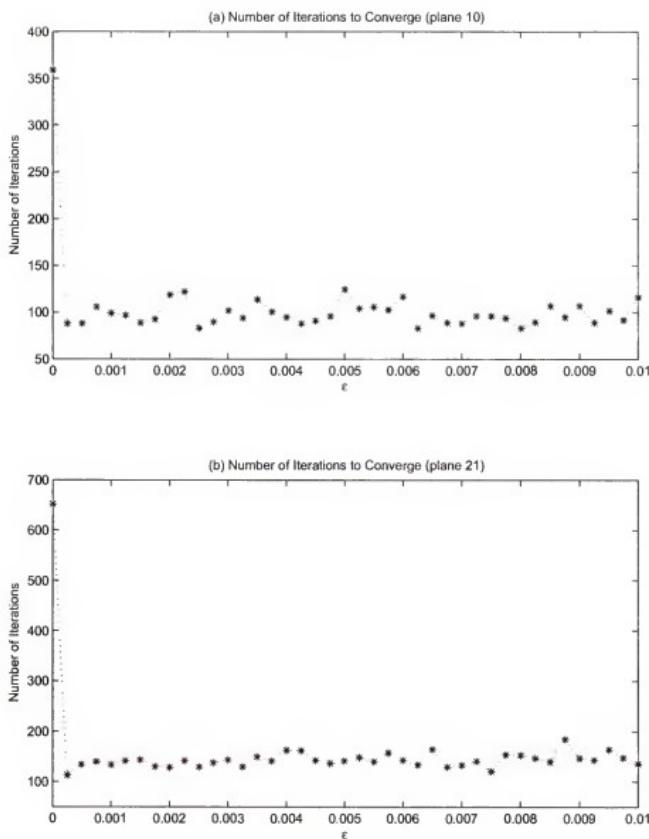


Figure 7.24: Number of iterations for the APML algorithm to decrease the PML objective function to $\Phi(\mathbf{x}^*)$ are shown for different values of ϵ when 14 minute data were used, where \mathbf{x}^* is the 5000th iterate of the PML algorithm.

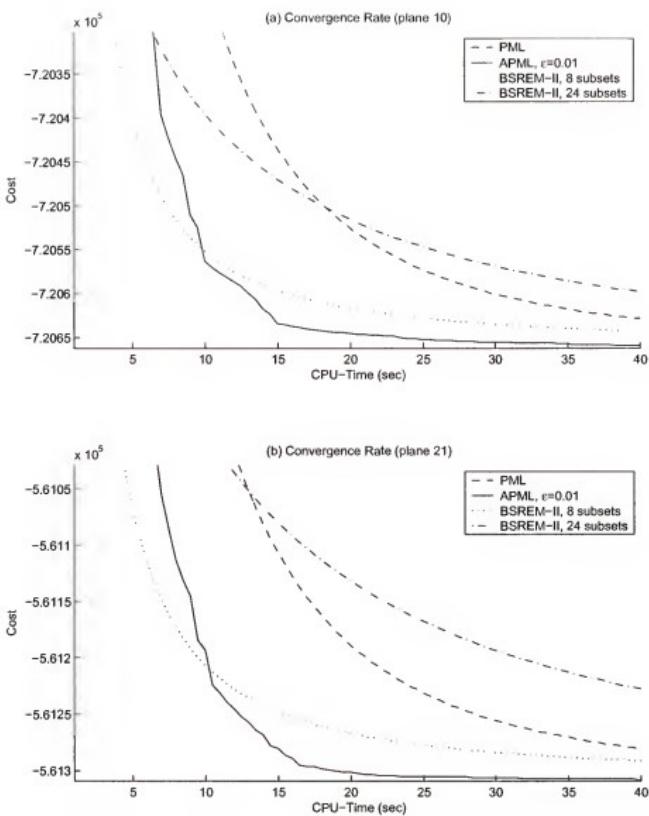


Figure 7-25: Convergence rate comparison of the PML, APML and the BSREM-II algorithms when 14 minute data were used.

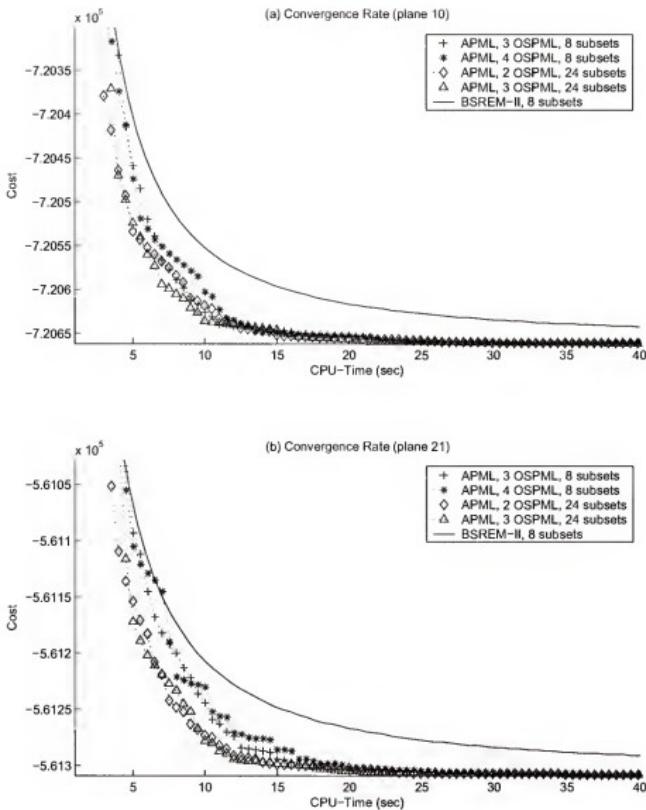


Figure 7-26: Convergence rate comparison of the APML algorithm with different number of OS-PML iterations for 8 and 24 subsets, and the BSREM-II algorithm for 8 subsets when 14 minute data were used. For the APML algorithm $\epsilon = 0.01$ was used.

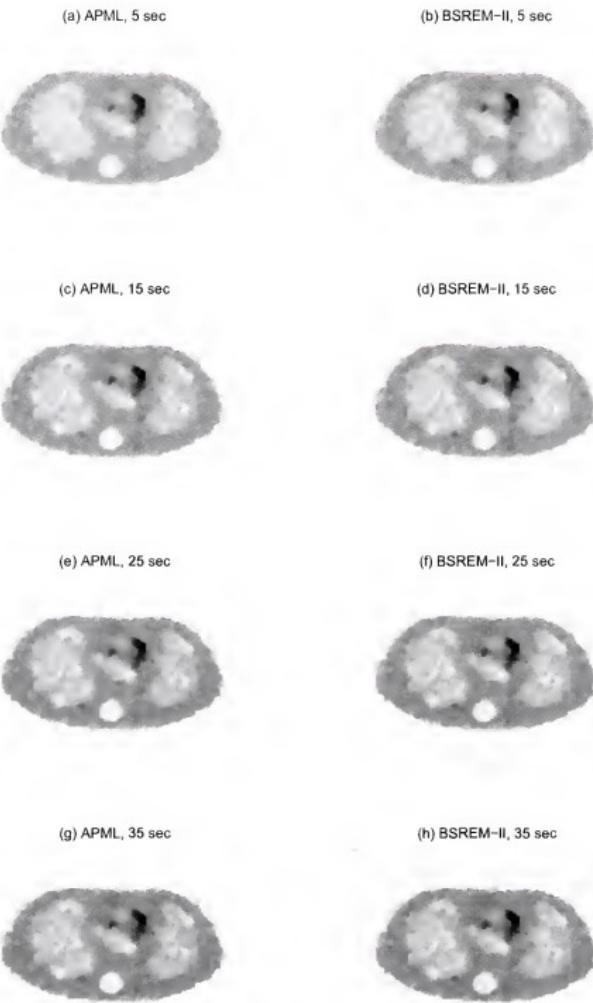


Figure 7-27: APML and BSREM-II iterates for plane 10 are shown for different CPU-times when 14 minute data was used. The APML iterates were generated by the APML algorithm with $\epsilon = 0.01$ and 2 OS-PML iterations for 24 subsets. The BSREM-II iterates were generated by the BSREM-II algorithm for 8 subsets. For display purposes, all the images were adjusted so that they have the same dynamic range.

(a) APML, 5 sec

(b) BSREM-II, 5 sec

(c) APML, 15 sec

(d) BSREM-II, 15 sec

(e) APML, 25 sec



(f) BSREM-II, 25 sec



(g) APML, 35 sec



(h) BSREM-II, 35 sec



Figure 7-28: APML and BSREM-II iterates for plane 21 are shown for different CPU-times when 14 minute data was used. The APML iterates were generated by the APML algorithm with $\epsilon = 0.01$ and 2 OS-PML iterations for 24 subsets. The BSREM-II iterates were generated by the BSREM-II algorithm for 8 subsets. For display purposes, all the images were adjusted so that they have the same dynamic range.

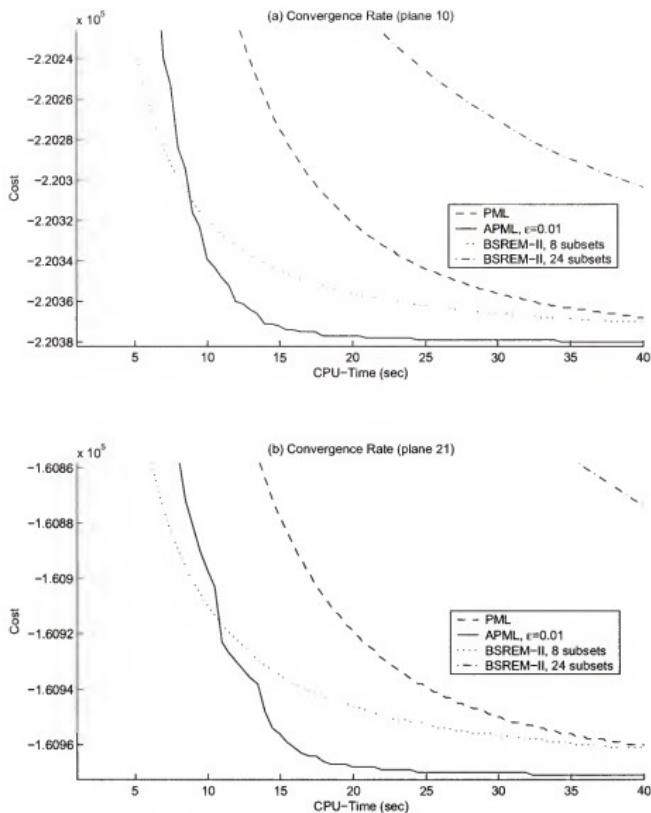


Figure 7-29: Convergence rate comparison of the PML, APML and the BSREM-II algorithms when 7 minute data were used.

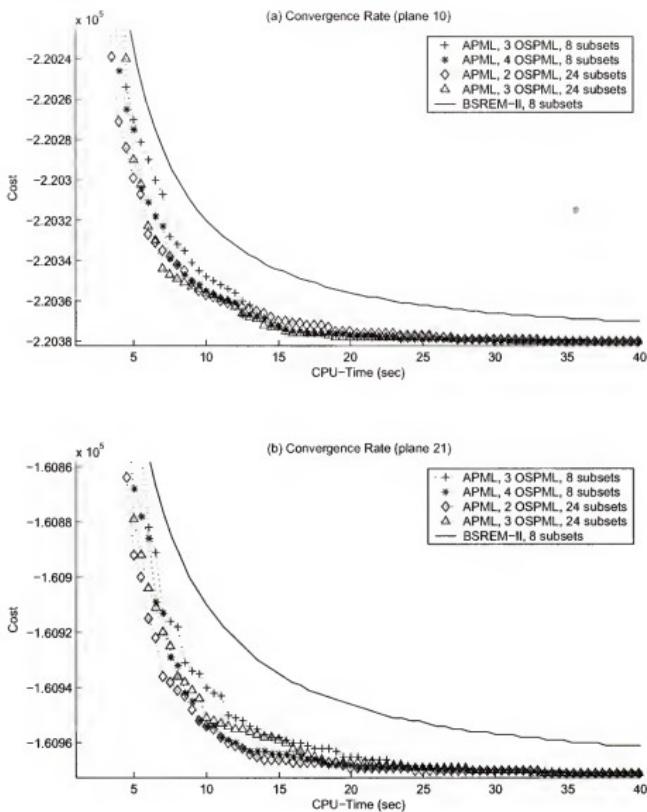


Figure 7-30: Convergence rate comparison of the APML algorithm with different number of OS-PML iterations for 8 and 24 subsets, and the BSREM-II algorithm for 8 subsets when 7 minute data were used. For the APML algorithm $\epsilon = 0.01$ was used.

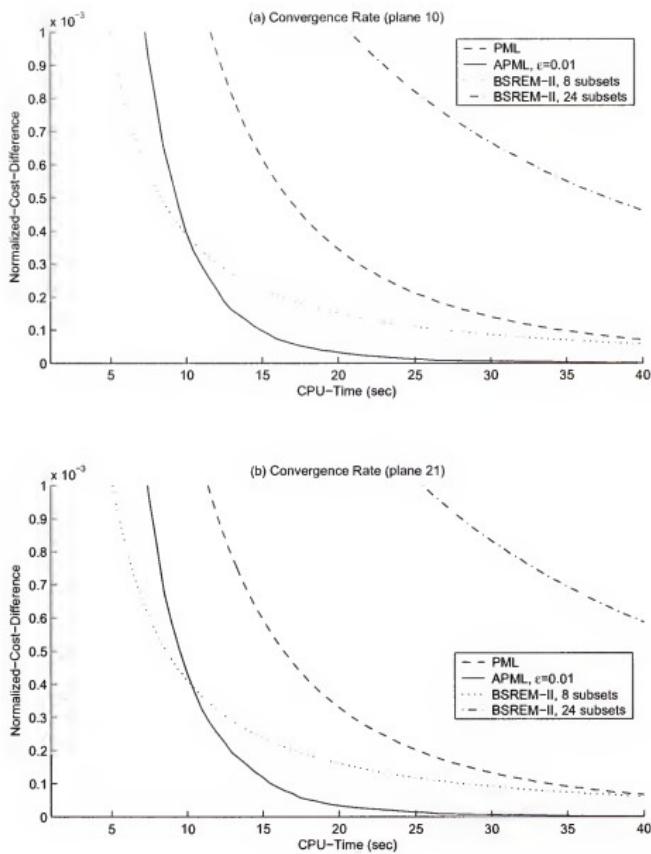


Figure 7-31: Average convergence rate comparison of the PML, APML and the BSREM-II algorithms when 7 minute data was used.

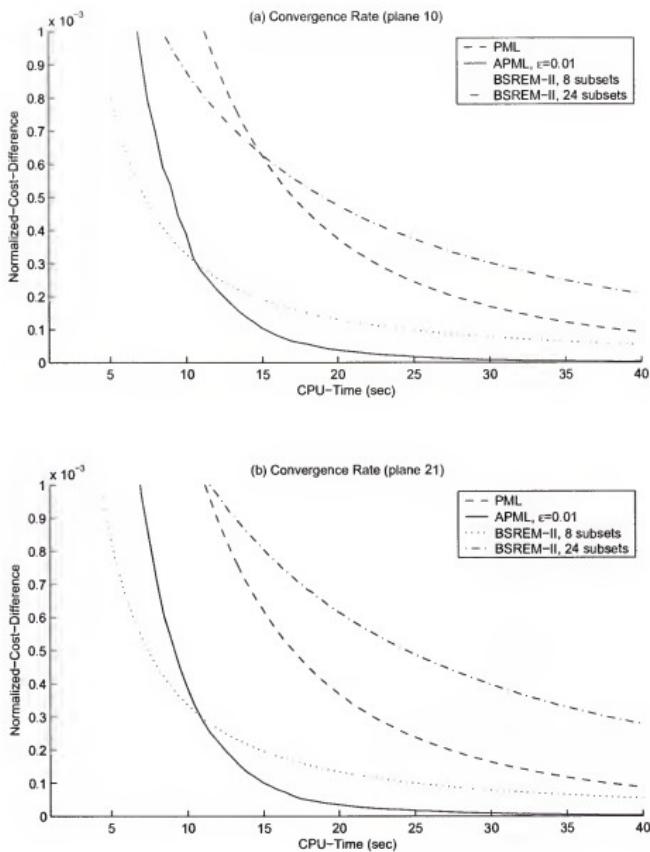


Figure 7-32: Average convergence rate comparison of the PML, APML and the BSREM-II algorithms when 14 minute data was used.

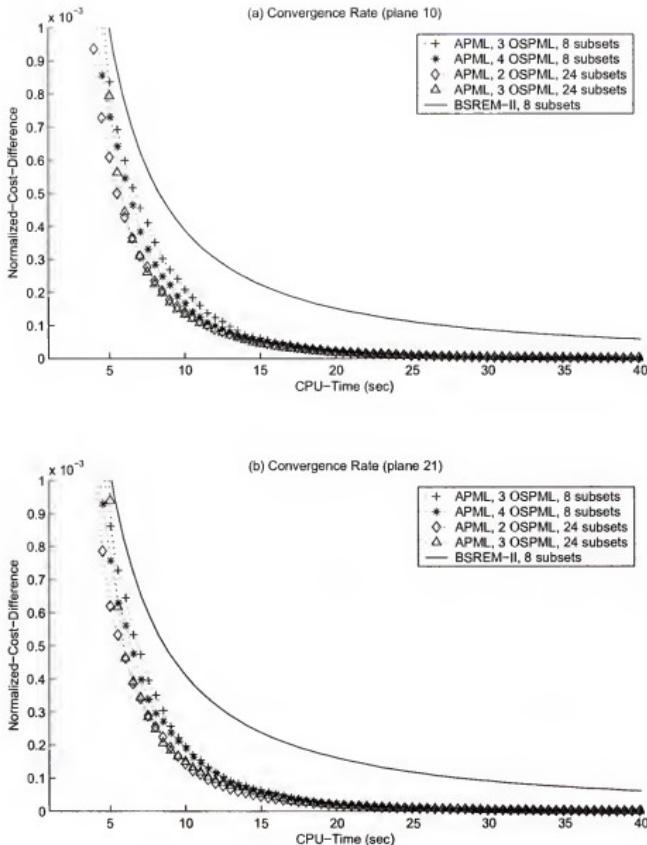


Figure 7-33: Average convergence rate comparison of the APML algorithm with different number of OS-PML iterations for 8 and 24 subsets, and the BSREM-II algorithm for 8 subsets when 7 minute data was used. For the APML algorithm $\epsilon = 0.01$ was used.

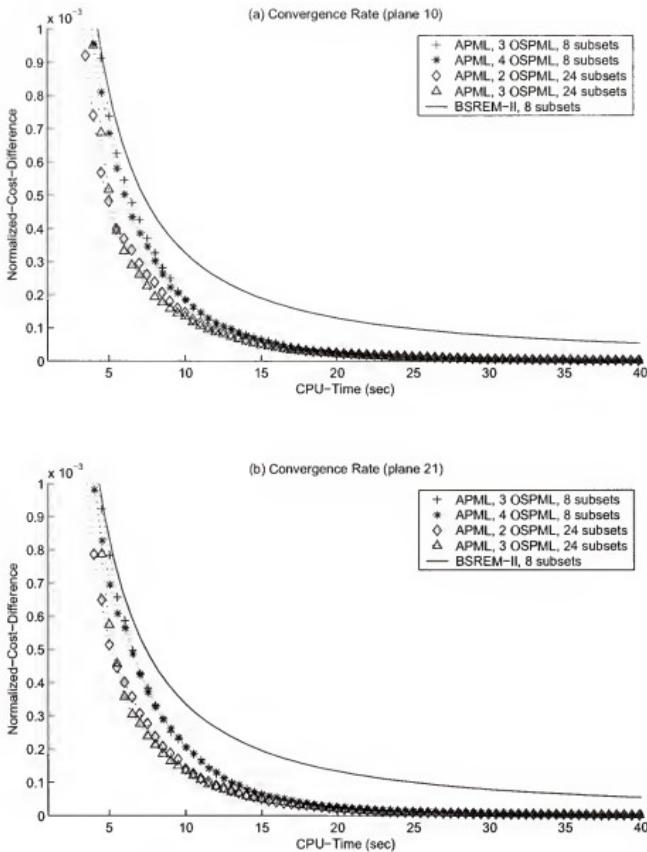


Figure 7-34: Average convergence rate comparison of the APML algorithm with different number of OS-PML iterations for 8 and 24 subsets, and the BSREM-II algorithm for 8 subsets when 14 minute data was used. For the APML algorithm $\epsilon = 0.01$ was used.

Table 7.1: Comparison of the computational complexity per iteration.

	PML	BSREM-II	APML
CPU-time (sec)	0.271565	0.58836 ($R = 8$)	0.49828
Memory Read	$2P$	$2P$	$3P$
Mul/Div	$2P + I + 24J$	$2P + I + 12RJ$	$4P + 8I + 58J$
Add/Sub	$2P + 31J$	$2P + I + 16RJ$	$4P + 3I + 66J$
\sqrt{t}	J	-	J
$\lambda(t)$	$8J$	$8RJ$	$16J$
N^c for plane 10	791	N/A	359 ($\epsilon = 0$), 116 ($\epsilon = 0.01$)
N^c for plane 21	1362	N/A	652 ($\epsilon = 0$), 136 ($\epsilon = 0.01$)

The letters P and R denote the number of non-zero elements in the probability matrix \mathcal{P} and the number of subsets for the BSREM-II algorithm, respectively. The letters I and J represent the number of data points and voxels, respectively. N^c denotes the number of iterations for convergence. All the algorithms were computed on a Dell Inspiron-5150 computer. Convergence in the table means that an algorithm decreases the PML objective function until it equals $\Phi(\mathbf{x}^*)$, where \mathbf{x}^* is the 5000th PML iterate.

probability matrix. In other words, given the MLEM image based on \mathcal{P}^c , the PCiPS algorithm first estimated the probability matrix, referred to as $\hat{\mathcal{P}}^{\text{true}}$, and then images were reconstructed by the APML algorithm with 100 iterations. In the PCiPS algorithm, two choices (15 and 63) for the parameter τ were examined in the experiments. We used the log-cosh function $\lambda(t) = \log(\cosh(\frac{t}{\delta}))$ with $\delta = 50$ as the penalty function.

The images in Figures 7.35 (a) and (b) were obtained by the MLEM-S and APML algorithms with \mathcal{P}^c . The images in Figures 7.35 (c) and (d) were obtained by the APML algorithm using $\hat{\mathcal{P}}^{\text{true}}$ with $\tau = 15$ and $\tau = 63$, respectively. For the APML images in Figure 7.35, β was chosen so that the background standard deviation is approximately same (background standard deviation is approximately 48). Using 5 iterations, the MLEM-S image with a background standard deviation of

48 was obtained. Figure 7 35 (a) illustrates that the tumors in the MLEM-S image is too smooth. Figures 7 35 (b), (c), and (d) show that the APML images visually look similar to each other. However, they are quite different especially around tumors. This claim is supported by Figure 7 36 that is a line plot (the row is shown in Figure 7 13) of the images in Figure 7 35. For the row under consideration, it can be seen from the line plots that the APML images in Figures 7 35 (c) and (d) have a higher degree of contrast than the other images. For the images in Figure 7 35, the large tumor contrast of the APML image in (d) was 59%, 17%, and 7% higher than the images in (a), (b), and (c), respectively. The increased contrast of the APML image in (d) for the small tumor with respect to the images in (a), (b), and (c) was 60%, 20%, and 14%, respectively. The increased tumor distinguishability of the APML image in (d) with respect to the images in (a), (b), and (c) was 30%, 10%, and 5%, respectively.

Figures 7 37 (a) and (b) are plots of the average contrast of the large tumor and small tumor versus the average background standard deviation using fifteen realizations for plane 21, respectively. Further, plots of the average standard deviation of the large tumor and the average distinguishability of two tumors versus the average background standard deviation for the fifteen realizations are shown in Figures 7 37 (c) and (d), respectively. The contrast and distinguishability curves of the APML algorithm with the estimated probability matrices lie above the corresponding curves of the other algorithms. Thus, for a fixed level of background noise, the APML images generated using the estimated probability matrices, on average, have the greatest contrast and distinguishability.

Now, we consider fifteen realizations of 7 minute real phantom data for plane 21. Figures 7 38 (a) and (b) are plots of the average contrast of the large tumor and small tumor versus the average background standard deviation using fifteen realizations, respectively. A plot of the average standard deviation of the large tumor versus the

average background standard deviation for the fifteen realizations is shown in Figure 7.38 (c). Also, in Figure 7.38 (d), we provide a plot of the average distinguishability of two tumors versus the average background standard deviation for fifteen realizations. As in the experiments with the 14 minute real phantom data, it is evident that the APML algorithm with the estimated probability matrices outperform the MLEM-S and APML algorithms with \mathcal{P}^c in terms of contrast recovery and tumor distinguishability.

Figures 7.39 (a) and (b) are the images obtained by the MLEM-S and APML algorithms using \mathcal{P}^c , respectively, for a 7 minute data. Figures 7.39 (c) and (d) are the images obtained by the APML algorithm using $\hat{\mathcal{P}}^{\text{true}}$ with $\tau = 15$ and $\tau = 63$, respectively, for a 7 minute data. Figure 7.39 (a) is the 7th MLEM iterate. For the APML algorithm, 100 iterations were used. For the images in Figures 7.39 (b), (c), and (d), β was chosen so that the standard deviation of their backgrounds are approximately 36. As in the experiments with the 14 minute data, the images in Figures 7.39 (b), (c), and (d) look similar to each other. Figure 7.40 is the line plot of the images in Figure 7.39. Also, as in the experiments with the 14 minute data, for the row under consideration, the APML images in Figures 7.39 (c) and (d) have a higher degree of contrast than the other images. For the images in Figure 7.39, the large tumor contrast of the APML image in (d) was 50%, 19%, and 6% higher than the images in (a), (b), and (c), respectively. The increased contrast of the APML image in (d) for the small tumor with respect to the images in (a), (b), and (c) was 47%, 19%, and 7%, respectively. The increased tumor distinguishability of the APML image in (d) with respect to the images in (a), (b), and (c) was 18%, 10%, and -1%, respectively.

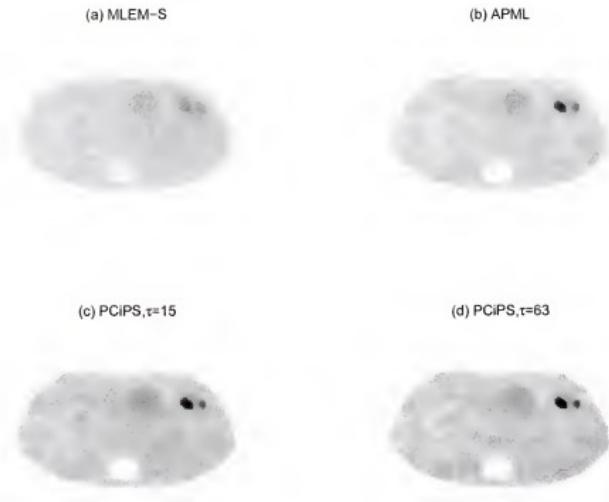


Figure 7-35: Comparison of emission images when a 14 minute real phantom data for plane 21 was used: (a) MLEM-S image with P^c , (b) APML image with P^c , (c) APML image with \hat{P}^{true} and $\tau = 15$, (d) APML image with \hat{P}^{true} and $\tau = 63$. The image in (a) is from 5 iterations, while the images in (b), (c), and (d) are from 100 iterations. For the images in (b), (c), and (d), β was chosen in such a way that the standard deviation of the background is approximately 48. Specifically, $\beta = 1/32$, $1/32$, and 0.028 for the images in (b), (c), and (d), respectively. For display purposes, all the images were adjusted so that they have the same dynamic range.

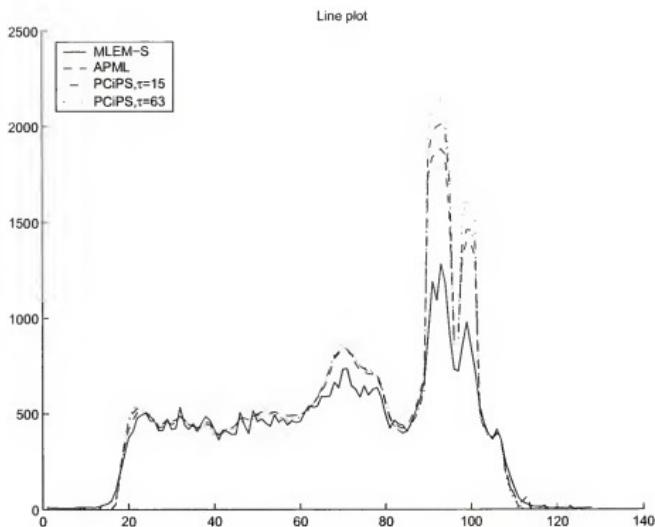


Figure 7-36: A line plot comparison of the reconstructed images in Figure 7-35.

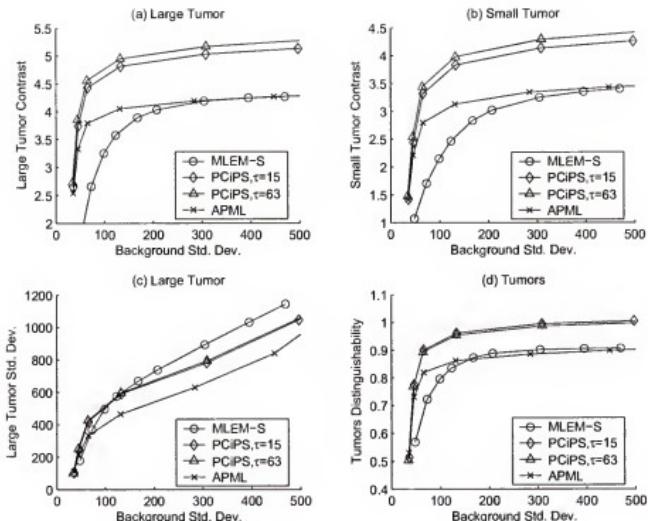


Figure 7-37: Plots of the average contrast of the large and small tumors versus the average background standard deviation are shown in (a) and (b), respectively. A plot of the average standard deviation of the large tumor versus the average background standard deviation is shown in (c). In (d), a plot of the average distinguishability of two tumors versus the average background standard deviation is shown. Fifteen realizations were used and 14 minute phantom data for plane 21 was used in the study. For the MLEM-S curves, the images from iterations 5 – 160 were used. For the other curves, the images were reconstructed using two hundred iterations for $\beta = 2^{-4} - 2^{-12}$.

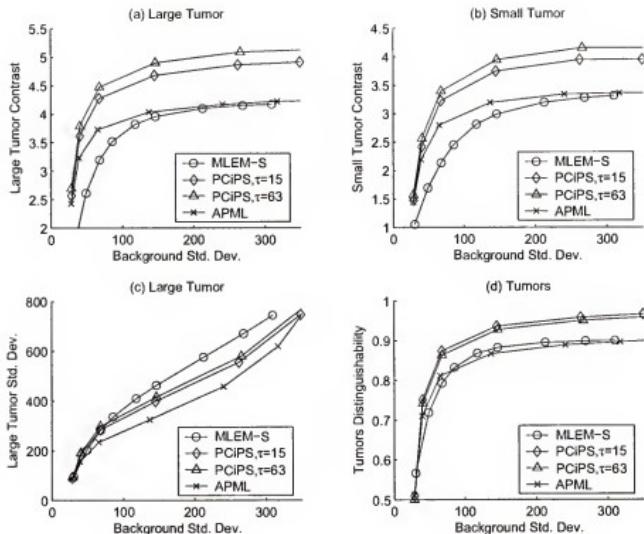


Figure 7-38: Plots of the average contrast of the large and small tumors versus the average background standard deviation are shown in (a) and (b), respectively. A plot of the average standard deviation of the large tumor versus the average background standard deviation is shown in (c). In (d), a plot of the average distinguishability of two tumors versus the average background standard deviation is shown. Fifteen realizations were used and 7 minute real phantom data for plane 21 was used in the study. For the MLEM-S curves, the images from iterations 5 – 160 were used. For the other curves, the images were reconstructed using two hundred iterations for $\beta = 2^{-4} - 2^{-12}$.

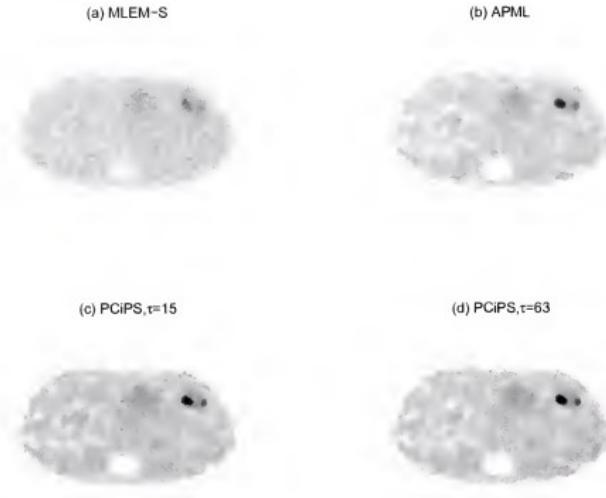


Figure 7-39: Comparison of emission images when a 7 minute real phantom data for plane 21 was used: (a) MLEM-S image with \mathcal{P}^c , (b) APML image with \mathcal{P}^c , (c) APML image with $\hat{\mathcal{P}}^{\text{true}}$ and $\tau = 15$, (d) APML image with $\hat{\mathcal{P}}^{\text{true}}$ and $\tau = 63$. The image in (a) is from 7 iterations, while the images in (b), (c), and (d) are from 100 iterations. For the images in (b), (c), and (d), β was chosen in such a way that the standard deviation of the background is approximately 36. Specifically, $\beta = 1/32$, $1/32$, and 0.028 for the images in (b), (c), and (d), respectively. For display purposes, all the images were adjusted so that they have the same dynamic range.

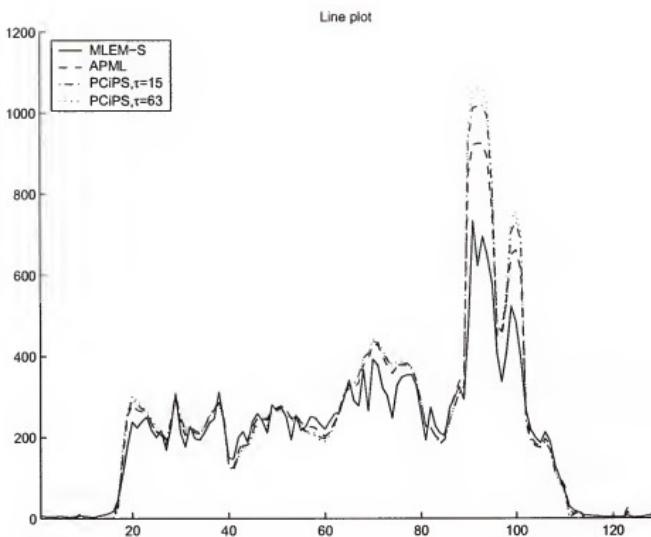


Figure 7-40: A line plot comparison of the reconstructed images in Figure 7-39.

CHAPTER 8 CONCLUSIONS AND FUTURE WORK

8.1 Conclusions

The PML algorithm we developed for reconstructing emission images generates nonnegative emission mean estimates and monotonically decreases the PML objective function. The algorithm is straightforward to implement and can incorporate any penalty function that satisfies the mild assumptions (AS3)-(AS8). Under certain conditions (i.e., (AS1)-(AS8)), the PML objective function is strictly convex. And, for the case where the PML objective function is strictly convex, it has been proven that the PML algorithm converges to the minimizer of the PML objective function. Although the tradeoff between resolution and noise can be controlled by certain regularization hyperparameters (i.e., β and δ), like many researchers, we have not determined a way to choose the parameters so that the data-fit and *a priori* knowledge are optimally “balanced”.

A fast version of the PML algorithm, called the APML algorithm, was developed that retains the properties of the PML algorithm. The APML algorithm is based on the PML algorithm and pattern search idea of Hooke and Jeeve. However, we modified the direction vector to account for the PET image reconstruction problem. The APML algorithm generates nonnegative emission mean estimates, monotonically decreases the PML objective function, and can accommodate the same class of penalty functions as the PML algorithm. Importantly, it has been proven that the APML algorithm converges to the minimizer of the PML objective function when the PML objective function is strictly convex. Although the APML algorithm requires an additional parameter ϵ that is used to define the direction vector, experiments using real phantom data demonstrated that fast convergence rates were obtained over a

wide range of values for ϵ (e.g., $0 < \epsilon \leq 0.01$). This means that the APML algorithm is robust with respect to the parameter ϵ .

In experiments using real phantom data, it was shown that the APML algorithm decreased the PML objective function about three times faster than the PML algorithm. Specifically, the APML algorithm for $0 < \epsilon \leq 0.01$ needed about one third of the CPU-time that was necessary for the PML algorithm to decrease the PML objective function to a “practical minimizer” of the objective function. By practical minimizer, we mean a PML iterate that has a PML objective value (i.e., cost) that, practically speaking, cannot be appreciably decreased with increasing iterations. Specifically, the minimum PML objective function value resulted from the 5000 iteration of the PML algorithm. We compared the convergence rate of the APML algorithm to an ordered-subsets method in [35] because the ordered-subsets based algorithm is said to converge to the nonnegative minimizer of the PML objective function. At the early iterations, the ordered-subsets method decreased the PML objective function at a faster rate than the APML algorithm. However, after about 10 seconds in CPU-time, the APML algorithm decreased faster. It was also shown that when the APML algorithm utilizes an ordered-subsets based PML iteration for a few early iterations, the resulting algorithm decreased the PML objective function more than about 1.2 times faster than the ordered-subsets method for the early iterations (i.e., less than 10 seconds in CPU-time).

In addition to the PML and APML algorithms, we also proposed a regularized image reconstruction algorithm we call the QEP algorithm. The QEP algorithm obtains regularized estimates of the emission means through the use of an iteration dependent penalty function that serves to preserve edges in the reconstructed images. The definition of the penalty function was motivated by an analysis of the surrogate function for a penalty function that is utilized by the PML algorithm. In the QEP algorithm, at each iteration, the iteration dependent penalty function is found and

the next iterate is originally defined to be a nonnegative minimizer of sum of the negative log likelihood function and the penalty function. However, since it is not possible to solve the constrained optimization problem, the QEP algorithm alternatively defines the next iterate to be a nonnegative minimizer of sum of the de-coupled surrogate function for the negative log likelihood function and the iteration dependent penalty function. It is important to understand that the QEP algorithm defines a new objective function to be minimized at each iteration. Thus, unlike the PML and APML algorithms, the QEP algorithm does not minimize a single objective function. This means that the usual mathematical tools for investigating convergence are unavailable. Despite its theoretical drawbacks, the QEP algorithm performed extremely well in experiments with computer-generated phantom data and real thorax phantom data, and outperformed the PML (or APML) algorithm and PWLS algorithm [42] in terms of contrast recovery.

In experiments, the images produced by the PML (or APML) and QEP algorithms had greater contrast and “distinguishability” than the MLEM-S and PWLS images for a fixed level of background noise. The MLEM-S images were produced by early termination of the MLEM algorithm [16] and the PWLS images were produced by the PWLS algorithm [42]. Specifically, for a 14 minute real phantom data set, the contrast of the large tumor in the QEP image was 37%, 3%, and 34% higher than the MLEM-S, PML, and PWLS images, respectively. With respect to the MLEM-S, PML, and PWLS images, the contrast of the small tumor in the QEP image was 46%, 9%, and 43% higher, respectively. The increased tumor distinguishability of the QEP image with respect to the MLEM-S, PML, and PWLS images was 20%, 5%, and 35%, respectively. Moreover, qualitatively speaking, the spatial extent of the tumors were more easily resolved with the PML and QEP images. Since the QEP algorithm yielded the greatest contrast for a fixed level of background noise, it may be particularly well suited for tumor detection applications.

Errors caused by scatter, non-collinearity, detector penetration, and positron range have the net effect of introducing blur into PET images. In theory, these errors can be corrected by determining a suitable probability matrix. However, it is difficult to determine such a probability matrix because the required modelling is impractical. In Chapter 6, we first assume that the “true” probability matrix for the observed emission data is a product of an unknown nonnegative matrix, called a scatter matrix, and a “conventional” probability matrix. The conventional probability matrix is generated from the geometry of the PET scanner and image space to be reconstructed, along with certain standard corrections for errors. We developed a method, referred to as the joint minimum Kullback-Leibler (KL) distance method, that aims to reduce blur in PET images. In the joint minimum KL distance method, the scatter matrix and emission means are jointly estimated by minimizing the distance the data and model parameters. Because of the difficulty of the minimization problem, the number of unknowns in the scatter matrix is reduced and an alternating minimization algorithm is developed. Thus, given an estimate for the scatter matrix, an estimate for the emission means is obtained. The estimate for the emission means can then be used to generate an improved estimate for the scatter matrix. Alternating between these two steps leads to the desired estimates for the scatter matrix and emission means. Once the estimate of the scatter matrix is obtained, the estimate for the true probability matrix is the a product of the estimated scatter matrix and conventional probability matrix. Then, a regularized image reconstruction algorithm is applied to the emission data using the estimated true probability matrix. In experiments with real phantom data, the APML algorithm is employed because of its fast convergence. The contrast of the reconstructed images generated using the estimated probability matrix was more accurate than the reconstructed images generated using the conventional probability matrix.

8.2 Future Work

Although the APML algorithm decreases the PML objective function faster than the PML algorithm in experiments, it is not clear that how fast the APML algorithm converges in theory. Consequently, it would be worthwhile to determine the theoretical rate of convergence of the APML algorithm. A related research direction would be to determine the parameter ϵ that maximizes convergence rate of the APML algorithm. Keeping in mind Hook and Jeeve's idea, it is possible that the convergence rate of the APML algorithm could be improved by using a "better" direction vector. We say this because, in experiments where the "best" direction vector $\mathbf{x}^{n+1} - \mathbf{x}^*$ was used (\mathbf{x}^* is the minimizer of the PML objective function) the APML algorithm converged in only a few iterations (e.g., 3 iterations).

A key assumption of the PCiPS algorithm is that the mean number of photon pairs recorded by the i^{th} detector pair is a weighted sum of the mean number of photon pairs that would have been recorded by a certain set of detector pairs if there were no errors due to scatter and non-collinearity. Currently, the chosen set of detector pairs is the projection for which the i^{th} detector pair belongs. In Chapter 6, the assumption was justified for the case where the image space had approximately uniform attenuation (e.g., brain). However, for an image space with non-uniform attenuation, another choice for the set of detector pairs may be more appropriate. Thus, it would be worthwhile to revisit the assumptions of the PCiPS algorithm so as to broaden its application.

Increasingly, three dimensional (3D) PET scanners are being used to perform whole-body scans. Thus, it may be beneficial to the PET community to extend the proposed algorithms to 3D. One practical problem of 3D PET is that it takes an extremely long time to reconstruct images because the number of data points is increased. So, with 3D implementations, one should consider parallel computing

techniques such as a multiprocessor approach [70] or a computer cluster implementation [71]. Observe that the proposed algorithms can be parallelized because all the pixel values are updated simultaneously. By contrast, the penalized weighted least-squares algorithm in [42] updates pixel values sequentially so that the algorithm cannot be parallelized.

The proposed algorithms were assessed with real phantom data. However, a more in-depth assessment would include other types of real phantoms (e.g., brain phantoms) and patient data. Another consideration is to study the proposed algorithms more when errors due to detector penetration, non-collinearity of line-of-response, and positron range are corrected. We did not consider those errors in the dissertation.

APPENDIX A HUBER'S SURROGATE FUNCTIONS

In this appendix, we present Huber's proof of the inequality $\bar{\lambda}^{(n)}(t) \geq \lambda(t)$ for all t . From (3.14), recall that $\bar{\lambda}^{(n)}$ is defined to be

$$\bar{\lambda}^{(n)}(t) = \lambda(t^{(n)}) + \dot{\lambda}(t^{(n)})(t - t^{(n)}) + \frac{1}{2}\gamma(t^{(n)})(t - t^{(n)})^2 \quad (\text{A.1})$$

$$= \lambda(t^{(n)}) + \dot{\lambda}(t^{(n)})(t - t^{(n)}) - \gamma(t^{(n)})tt^{(n)} + \frac{1}{2}\gamma(t^{(n)})(t^2 + (t^{(n)})^2) \quad (\text{A.2})$$

$$= \lambda(t^{(n)}) + \dot{\lambda}(t^{(n)})(t - t^{(n)}) - \dot{\lambda}(t^{(n)})t + \frac{1}{2}\dot{\lambda}(t^{(n)})t^{(n)} + \frac{1}{2}\gamma(t^{(n)})t^2 \quad (\text{A.3})$$

$$= \lambda(t^{(n)}) - \frac{1}{2}\dot{\lambda}(t^{(n)})t^{(n)} + \frac{1}{2}\gamma(t^{(n)})t^2, \quad (\text{A.4})$$

where we used that $\gamma(t) \triangleq \frac{\dot{\lambda}(t)}{t}$. Consider the function $f(t) \triangleq \bar{\lambda}^{(n)}(t) - \lambda(t)$ and its first derivative

$$\dot{f}(t) = [\gamma(t^{(n)}) - \gamma(t)]t. \quad (\text{A.5})$$

From (A.5) and the assumption that $\gamma(t)$ is nonincreasing over $[0, \infty)$ (see **(AS6)**), it follows that $\dot{f}(t) \leq 0$ over $[0, t^{(n)}]$ and $\dot{f}(t) \geq 0$ over $[t^{(n)}, \infty)$. These inequalities and the fact that $f(t^{(n)}) = 0$ imply that $f(t) \geq 0$ for $t \geq 0$, or equivalently $\bar{\lambda}^{(n)}(t) \geq \lambda(t)$ for $t \geq 0$. It is clear that $\bar{\lambda}^{(n)}(t) \geq \lambda(t)$ for $t \leq 0$ because of the symmetry of $\bar{\lambda}^{(n)}(t)$ and $\lambda(t)$ (see **(AS3)**).

APPENDIX B SURROGATE FUNCTIONS FOR PENALTY FUNCTION

Our objective in this appendix is to demonstrate that the surrogate function $\Lambda^{(n)}$ can be expressed as (3.22). First, we make the following observation: Since $\gamma(t) \triangleq \frac{\dot{\lambda}(t)}{t}$ (see (AS6)), $g^{(n)}$ in (3.17) can be written as

$$\begin{aligned} g^{(n)}(x_j, x_k) &= \lambda(x_j^{(n)} - x_k^{(n)}) + \dot{\lambda}(x_j^{(n)} - x_k^{(n)})[(x_j - x_j^{(n)}) - (x_k - x_k^{(n)})] \\ &\quad + \frac{1}{4}\gamma(x_j^{(n)} - x_k^{(n)})[(2x_j - 2x_j^{(n)})^2 + (2x_k - 2x_k^{(n)})^2] \end{aligned} \quad (\text{B.1})$$

$$\begin{aligned} &= \lambda(x_j^{(n)} - x_k^{(n)}) + \gamma(x_j^{(n)} - x_k^{(n)})[(x_j - x_j^{(n)})^2 + (x_k - x_k^{(n)})^2] \\ &\quad + \gamma(x_j^{(n)} - x_k^{(n)})(x_j^{(n)} - x_k^{(n)})[(x_j - x_j^{(n)}) - (x_k - x_k^{(n)})] \end{aligned} \quad (\text{B.2})$$

$$\begin{aligned} &= \lambda(x_j^{(n)} - x_k^{(n)}) + \gamma(x_j^{(n)} - x_k^{(n)})[x_j^2 - 2x_j x_j^{(n)} + (x_j^{(n)})^2 \\ &\quad + x_k^2 - 2x_k x_k^{(n)} + (x_k^{(n)})^2] + \gamma(x_j^{(n)} - x_k^{(n)})[x_j^{(n)} x_j - x_k^{(n)} x_j \\ &\quad - (x_j^{(n)})^2 + x_k^{(n)} x_j^{(n)} - x_j^{(n)} x_k + x_k^{(n)} x_k + x_j^{(n)} x_k^{(n)} - (x_k^{(n)})^2] \end{aligned} \quad (\text{B.3})$$

$$\begin{aligned} &= \gamma(x_j^{(n)} - x_k^{(n)})[x_j^2 - (x_j^{(n)} + x_k^{(n)}) x_j + x_k^2 - (x_j^{(n)} + x_k^{(n)}) x_k] \\ &\quad + 2\gamma(x_j^{(n)} - x_k^{(n)}) x_j^{(n)} x_k^{(n)} + \lambda(x_j^{(n)} - x_k^{(n)}) \end{aligned} \quad (\text{B.4})$$

$$\begin{aligned} &= \gamma(x_j^{(n)} - x_k^{(n)}) \left[\left\{ x_j - \frac{x_j^{(n)} + x_k^{(n)}}{2} \right\}^2 + \left\{ x_k - \frac{x_k^{(n)} + x_j^{(n)}}{2} \right\}^2 \right] \\ &\quad - \gamma(x_j^{(n)} - x_k^{(n)}) \frac{(x_j^{(n)} + x_k^{(n)})^2}{2} + 2\gamma(x_j^{(n)} - x_k^{(n)}) x_j^{(n)} x_k^{(n)} \\ &\quad + \lambda(x_j^{(n)} - x_k^{(n)}) \end{aligned} \quad (\text{B.5})$$

$$\begin{aligned} &= \gamma(x_j^{(n)} - x_k^{(n)}) \left[\left\{ x_j - \frac{x_j^{(n)} + x_k^{(n)}}{2} \right\}^2 + \left\{ x_k - \frac{x_k^{(n)} + x_j^{(n)}}{2} \right\}^2 \right] \\ &\quad - \frac{1}{2}\gamma(x_j^{(n)} - x_k^{(n)})(x_j^{(n)} - x_k^{(n)})^2 + \lambda(x_j^{(n)} - x_k^{(n)}) \end{aligned} \quad (\text{B.6})$$

$$\begin{aligned} &= \gamma(x_j^{(n)} - x_k^{(n)}) \left[\left\{ x_j - \frac{x_j^{(n)} + x_k^{(n)}}{2} \right\}^2 + \left\{ x_k - \frac{x_k^{(n)} + x_j^{(n)}}{2} \right\}^2 \right] \\ &\quad + C_{jk}^{(n)}, \end{aligned} \quad (\text{B.7})$$

where $C_{jk}^{(n)} = \lambda(x_j^{(n)} - x_k^{(n)}) - \frac{1}{2}\dot{\lambda}(x_j^{(n)} - x_k^{(n)})(x_j^{(n)} - x_k^{(n)})$. Thus, $\Lambda^{(n)}$ in (3.18) can be written as

$$\Lambda^{(n)}(\boldsymbol{x}) = 2 \sum_{j=1}^J \sum_{k \in N_j} \omega_{jk} \gamma(x_j^{(n)} - x_k^{(n)}) \left\{ x_j - \frac{x_j^{(n)} + x_k^{(n)}}{2} \right\}^2 + C^{(n)}, \quad (\text{B.8})$$

where $C^{(n)} = \sum_{j=1}^J \sum_{k \in N_j} \omega_{jk} c_{jk}^{(n)}$. To get (B.8), we used the fact that the function γ is even symmetric (i.e., $\gamma(t) = \gamma(-t)$ for all t) and $\omega_{jk} = w_{kj}$ for all j and k . We also used the fact that the j^{th} voxel is excluded from the set N_j , and, if the k^{th} voxel is in N_j , then the j^{th} voxel is in N_k .

APPENDIX C STRICT CONVEXITY OF PML OBJECTIVE FUNCTION

We will prove that the PML objective function Φ is strictly convex over the set $\{\mathbf{z} : \mathbf{z} \geq \mathbf{0}\}$. First, we will show that the Hessian of Λ , denoted by \mathcal{S} , satisfies the following properties:

- (S1) $\mathbf{z}' \mathcal{S} \mathbf{z} \geq 0$ for all \mathbf{z} , where $'$ denotes the transpose operator
- (S2) $\mathbf{z}' \mathcal{S} \mathbf{z} = 0$ only when $\mathbf{z} = \mathbf{0}$ or $\mathbf{z} = c\mathbf{1}$ for some $c \neq 0$.

The $(l, m)^{\text{th}}$ element of \mathcal{S} is given by

$$[\mathcal{S}]_{lm} = \begin{cases} 2 \sum_{k \in N_l} w_{lk} \ddot{\lambda}(x_l - x_k), & l = m \\ -2w_{lm} \ddot{\lambda}(x_l - x_m), & l \neq m \text{ and } m \in N_l \\ 0, & \text{otherwise} \end{cases}. \quad (\text{C.1})$$

Using (C.1), it follows that

$$\mathbf{z}' \mathcal{S} \mathbf{z} = \sum_{l=1}^J z_l \left\{ 2z_l \sum_{m \in N_l} w_{lm} \ddot{\lambda}(x_l - x_m) - 2 \sum_{m \in N_l} z_m w_{lm} \ddot{\lambda}(x_l - x_m) \right\} \quad (\text{C.2})$$

$$= 2 \sum_{l=1}^J \sum_{m \in N_l} w_{lm} \ddot{\lambda}(x_l - x_m) z_l (z_l - z_m) \quad (\text{C.3})$$

$$= 2 \sum_{l=1}^J \sum_{m \in N_l, m > l} w_{lm} \ddot{\lambda}(x_l - x_m) (z_l - z_m)^2. \quad (\text{C.4})$$

Since $w_{lm} \ddot{\lambda}(x_l - x_m) > 0$ for all l and m (see (AS5)), (S1) and (S2) follow from (C.4).

To prove that Φ is strictly convex, it suffices to show that $\mathbf{z}' \mathcal{T} \mathbf{z} > 0$ for $\mathbf{z} \neq \mathbf{0}$, where \mathcal{T} is the Hessian of Φ . Note that $\mathcal{T} = \mathcal{R} + \beta \mathcal{S}$, where \mathcal{R} is the Hessian of $-\Psi$.

The $(l, m)^{th}$ element of \mathcal{R} equals

$$[\mathcal{R}]_{lm} = \sum_{i=1}^I d_i \frac{\mathcal{P}_{il} \mathcal{P}_{im}}{[\mathcal{P}\mathbf{x} + \boldsymbol{\rho}]_i^2}. \quad (\text{C.5})$$

By **(S1)** and the fact that

$$\mathbf{z}' \mathcal{R} \mathbf{z} = \sum_{i=1}^I \frac{d_i}{[\mathcal{P}\mathbf{x} + \boldsymbol{\rho}]_i^2} \left(\sum_{m=1}^J \mathcal{P}_{im} z_m \right) \left(\sum_{l=1}^J \mathcal{P}_{il} z_l \right) \quad (\text{C.6})$$

$$= \sum_{i=1}^I d_i \left(\frac{[\mathcal{P}\mathbf{z}]_i}{[\mathcal{P}\mathbf{x} + \boldsymbol{\rho}]_i} \right)^2, \quad (\text{C.7})$$

it is true that $\mathbf{z}' T \mathbf{z} \geq 0$ for all \mathbf{z} . By **(AS1)** and **(AS2)**, it follows that

$$\mathbf{z}' \mathcal{R} \mathbf{z} = c^2 \sum_{i=1}^I d_i \left(\frac{[\mathcal{P}\mathbf{1}]_i}{[\mathcal{P}\mathbf{x} + \boldsymbol{\rho}]_i} \right)^2 > 0 \quad (\text{C.8})$$

for $\mathbf{z} = c\mathbf{1}$ when $c \neq 0$. Thus, by **(S2)**, it is concluded that $\mathbf{z}' T \mathbf{z} > 0$ for $\mathbf{z} \neq \mathbf{0}$.

APPENDIX D
SOLUTION TO UNCONSTRAINED OPTIMIZATION PROBLEM
IN MODIFIED APML LINE SEARCH

We first show in this appendix that the surrogate function $\Gamma^{(n+1)}$ in (4.29) is strictly convex. Then, we prove that the solution to the unconstrained optimization problem in (4.30) can be expressed by (4.31). Consider the second derivative of $\Gamma^{(n+1)}$:

$$\ddot{\Gamma}^{(n+1)} = \sum_{i=1}^I \mu_i^{(n+1)} + \beta \sum_{j=1}^J \sum_{k \in N_j} \omega_{jk} \gamma(x_j^{(n+1)} - x_k^{(n+1)}) (v_j^{(n+1)} - v_k^{(n+1)})^2 \quad (\text{D.1})$$

(observe: $\ddot{\Gamma}^{(n+1)}$ is independent of α). It is true that $\{\mathbf{x}^{(n)}\}$ is bounded for all n , $\gamma(t) > 0$ for $-\infty < t < \infty$, and γ is a continuous function over $-\infty < t < \infty$. Thus, it is clear that the second term in the right hand side of (D.1) is positive for $\mathbf{v}^{(n+1)} \neq c\mathbf{1}$ provided $c \neq 0$. For the case where $\mathbf{v}^{(n+1)} = c\mathbf{1}$ with $c \neq 0$, it follows that from (4.27)

$$\mu_i^{(n+1)} = \begin{cases} \frac{c^2 d_i([\mathcal{P}\mathbf{1}]_i)^2}{(c[\mathcal{P}\mathbf{1}]_i L^{(n+1)} + [\mathcal{P}\mathbf{x}^{(n+1)}]_i + \rho_i)^2}, & c > 0 \\ \frac{c^2 d_i([\mathcal{P}\mathbf{1}]_i)^2}{(c[\mathcal{P}\mathbf{1}]_i U^{(n+1)} + [\mathcal{P}\mathbf{x}^{(n+1)}]_i + \rho_i)^2}, & c < 0 \end{cases} \quad (\text{D.2})$$

for all i . Thus, by (AS1) and (AS2), $\sum_{i=1}^I \mu_i^{(n+1)} > 0$ for $\mathbf{v}^{(n+1)} = c\mathbf{1}$, when $c \neq 0$. Consequently, it can be concluded that $\ddot{\Gamma}^{(n+1)} > 0$ for $\mathbf{v}^{(n+1)} \neq \mathbf{0}$. This result implies that $\Gamma^{(n+1)}$ is strictly convex for $\mathbf{v}^{(n+1)} \neq \mathbf{0}$.

Now, we will show that the expression in (4.31) is the solution to the unconstrained optimization problem in (4.30). First, recall that

$$\bar{\Lambda}^{(n+1)}(\mathbf{x}) = \sum_{j=1}^J \sum_{k \in N_j} \omega_{jk} \bar{\lambda}^{(n+1)}(x_j - x_k), \quad (\text{D.3})$$

where

$$\bar{\lambda}^{(n+1)}(t) = \lambda(t^{(n+1)}) + \dot{\lambda}(t^{(n+1)})(t - t^{(n+1)}) + \frac{1}{2}\gamma(t^{(n+1)})(t - t^{(n+1)})^2. \quad (\text{D.4})$$

For $t^{(n+1)} = x_j^{(n+1)} - x_k^{(n+1)}$, the surrogate function $\Gamma^{(n+1)}$ can be expressed as

$$\Gamma^{(n+1)}(\alpha) = \Theta^{(n+1)}(\alpha) + \beta\bar{\Lambda}^{(n+1)}(\mathbf{x}^{(n+1)} + \alpha\mathbf{v}^{(n+1)}) \quad (\text{D.5})$$

$$\begin{aligned} &= \sum_{i=1}^I \theta_i^{(n+1)}(\alpha) + \beta \sum_{j=1}^J \sum_{k \in N_j} \omega_{jk} \lambda(x_j^{(n+1)} - x_k^{(n+1)}) + C \\ &\quad + \beta \sum_{j=1}^J \sum_{k \in N_j} \omega_{jk} \dot{\lambda}(x_j^{(n+1)} - x_k^{(n+1)}) (v_j^{(n+1)} - v_k^{(n+1)}) \alpha \\ &\quad + \beta \sum_{j=1}^J \sum_{k \in N_j} \omega_{jk} \frac{1}{2} \gamma(x_j^{(n+1)} - x_k^{(n+1)}) (v_j^{(n+1)} - v_k^{(n+1)})^2 \alpha^2, \end{aligned} \quad (\text{D.6})$$

where $C \triangleq \sum_{i=1}^I \{\rho_i + \log(d_i!)\}$. Thus, using the definition of $\theta_i^{(n+1)}$ in (4.23), the first derivative of $\Gamma^{(n+1)}$, denoted by $\dot{\Gamma}^{(n+1)}$, is

$$\begin{aligned} \dot{\Gamma}^{(n+1)}(\alpha) &= \sum_{i=1}^I \{\mu_i^{(n+1)} \alpha + \dot{\psi}_i^{(n+1)}(0)\} \\ &\quad + \beta \sum_{j=1}^J \sum_{k \in N_j} \omega_{jk} \dot{\lambda}(x_j^{(n+1)} - x_k^{(n+1)}) (v_j^{(n+1)} - v_k^{(n+1)}) \\ &\quad + \beta \sum_{j=1}^J \sum_{k \in N_j} \omega_{jk} \gamma(x_j^{(n+1)} - x_k^{(n+1)}) (v_j^{(n+1)} - v_k^{(n+1)})^2 \alpha. \end{aligned} \quad (\text{D.7})$$

Since $\Gamma^{(n+1)}$ is strictly convex, the minimizer of $\Gamma^{(n+1)}$ can be found by setting its first derivative to zero:

$$\dot{\Gamma}^{(n+1)}(\alpha) \Big|_{\alpha=\bar{\alpha}^{(n+1)}} = 0. \quad (\text{D.8})$$

Solving (D.8) results in the expression in (4.31).

APPENDIX E
CONVEXITY OF SURROGATE FUNCTIONS
FOR OBJECTIVE FUNCTIONS IN APML LINE SEARCH

In this appendix, our objective is to show that there exists a symmetric positive definite matrix \mathcal{M} , which is independent of n , such that $\ddot{\Gamma}^{(n+1)} \geq 2(\mathbf{v}^{(n+1)})' \mathcal{M}(\mathbf{v}^{(n+1)})$, where $\ddot{\Gamma}^{(n+1)}$ is the second derivative of the surrogate function $\Gamma^{(n+1)}$ in (4.29). Note that $\ddot{\Gamma}^{(n+1)}$ is shown in (D.1). It is true that $\{\mathbf{x}^{(n)}\}$ is bounded for all n , $\gamma(t) > 0$ for or $-\infty < t < \infty$, and γ is a continuous function over $-\infty < t < \infty$. Thus, there exists a number $\gamma_0 > 0$ such that $\gamma(x_j^{(n+1)} - x_k^{(n+1)}) \geq \gamma_0$ for all j, k , and n . Hence, it follows that

$$\ddot{\Gamma}^{(n+1)} \geq \sum_{i=1}^I \mu_i^{(n+1)} + \beta \gamma_0 \sum_{j=1}^J \sum_{k \in N_j} \omega_{jk} (v_j^{(n+1)} - v_k^{(n+1)})^2 \quad (\text{E.1})$$

$$\geq \sum_{i=1}^I \mu_i^{(n+1)} + 2\beta \gamma_0 (\mathbf{v}^{(n+1)})' \mathcal{H}(\mathbf{v}^{(n+1)}) , \quad (\text{E.2})$$

where the $(l, m)^{th}$ element of \mathcal{H} is given by

$$[\mathcal{H}]_{lm} = \begin{cases} \sum_{k \in N_l} w_{lk}, & l = m \\ -w_{lm}, & l \neq m \text{ and } m \in N_l \\ 0, & \text{otherwise} \end{cases} . \quad (\text{E.3})$$

Note that the matrix \mathcal{H} is symmetric because $w_{lm} = w_{ml}$ for all l and m . Now, we consider the term $\sum_i \mu_i^{(n+1)}$. First, note that $\mu_i^{(n+1)}$ can be expressed as (see (4.27))

$$\mu_i^{(n+1)} = \begin{cases} f(s_i^{(n+1)}) d_i ([\mathcal{P} \mathbf{v}^{(n+1)}]_i)^2, & [\mathcal{P} \mathbf{v}^{(n+1)}]_i \neq 0 \\ 0, & [\mathcal{P} \mathbf{v}^{(n+1)}]_i = 0 , \end{cases} . \quad (\text{E.4})$$

where $f(t) \triangleq \frac{1}{t^2}$ and

$$s_i^{(n+1)} \triangleq \begin{cases} [\mathcal{P}\mathbf{v}^{(n+1)}]_i L^{(n+1)} + [\mathcal{P}\mathbf{x}^{(n+1)}]_i + \rho_i, & [\mathcal{P}\mathbf{v}^{(n+1)}]_i > 0 \\ [\mathcal{P}\mathbf{v}^{(n+1)}]_i U^{(n+1)} + [\mathcal{P}\mathbf{x}^{(n+1)}]_i + \rho_i, & [\mathcal{P}\mathbf{v}^{(n+1)}]_i < 0 \end{cases}. \quad (\text{E.5})$$

Also, note that $s_i^{(n+1)}$ is bounded below by 0 for all n and i (see (4.15)). Moreover, if we assume that $\mathbf{v}^{(n+1)}$ is bounded for all n , it follows that $s_i^{(n+1)}$ is bounded above for all n and i . To see this fact, we only need to consider $L^{(n+1)}$ and $U^{(n+1)}$ because $\mathbf{x}^{(n+1)}$ is bounded for all n by *Proposition 1*. When $v_j^{(n+1)} \leq 0$ for all j , it follows that $L^{(n+1)} = -\infty$. For this case, the fact that $L^{(n+1)} = -\infty$ is not a problem because $s_i^{(n+1)} = [\mathcal{P}\mathbf{v}^{(n+1)}]_i U^{(n+1)} + [\mathcal{P}\mathbf{x}^{(n+1)}]_i + \rho_i$, where $U^{(n+1)}$ is a finite number. Similarly, when $v_j^{(n+1)} \geq 0$ for all j , it follows that $U^{(n+1)} = \infty$. For this case, $s_i^{(n+1)} = [\mathcal{P}\mathbf{v}^{(n+1)}]_i L^{(n+1)} + [\mathcal{P}\mathbf{x}^{(n+1)}]_i + \rho_i$, where $L^{(n+1)}$ is a finite number. Thus, $s_i^{(n+1)}$ is bounded for all n and i . Since $f(t) > 0$ for $0 < t < \infty$, the function f is a continuous function over $0 < t < \infty$, and $s_i^{(n+1)} > 0$ is bounded above, there exists a number $f_0 > 0$ such that $f(s_i^{(n+1)}) \geq f_0$ for all n and i . Thus, it follows that

$$\ddot{\mathbf{f}}^{(n+1)} \geq f_0 \sum_{i=1}^I d_i ([\mathcal{P}\mathbf{v}^{(n+1)}]_i)^2 + 2\beta\gamma_0(\mathbf{v}^{(n+1)})' \mathcal{H}(\mathbf{v}^{(n+1)}) \quad (\text{E.6})$$

$$\geq f_0(\mathbf{v}^{(n+1)})' \mathcal{P}' \mathcal{D} \mathcal{P}(\mathbf{v}^{(n+1)}) + 2\beta\gamma_0(\mathbf{v}^{(n+1)})' \mathcal{H}(\mathbf{v}^{(n+1)}) \quad (\text{E.7})$$

$$\geq 2(\mathbf{v}^{(n+1)})' \mathcal{M}(\mathbf{v}^{(n+1)}), \quad (\text{E.8})$$

where $\mathcal{D} = \text{diag}(\mathbf{d})$ and $\mathcal{M} = \left\{ \frac{f_0}{2} \mathcal{P}' \mathcal{D} \mathcal{P} + \beta\gamma_0 \mathcal{H} \right\}$. Since $\mathcal{P}' \mathcal{D} \mathcal{P}$ and \mathcal{H} are symmetric, the matrix \mathcal{M} is symmetric. Now, we only need to show that \mathcal{M} is positive definite. From the second term in the right hand side of (E.1), it is easy to see that $\mathbf{z}' \mathcal{H} \mathbf{z} \geq 0$ for all \mathbf{z} and $\mathbf{z}' \mathcal{H} \mathbf{z} = 0$ only when $\mathbf{z} = \mathbf{0}$ or $\mathbf{z} = c\mathbf{1}$ for some $c \neq 0$. Moreover, by **(AS1)** and **(AS2)**, it follows that

$$\mathbf{z}' \mathcal{P}' \mathcal{D} \mathcal{P} \mathbf{z} = c^2 \sum_{i=1}^I d_i ([\mathcal{P}\mathbf{1}]_i)^2 > 0 \quad (\text{E.9})$$

for $\mathbf{z} = c\mathbf{1}$ and $c \neq 0$. Thus, from $f_0 > 0$ and $\beta\gamma_0 > 0$, it can be concluded that \mathcal{M} is positive definite.

REFERENCES

- [1] E. V. Garcia, T. L. Faber, J. R. Galt, C. D. Cooke, and R. D. Folks, "Advances in nuclear emission PET and SPECT imaging," *IEEE Eng. Med. Biol. Mag.*, vol. 19, pp. 21–33, Sept./Oct. 2000.
- [2] M. M. Maisey, R. L. Wahl, and S. F. Barrington, *Atlas of clinical positron emission tomography*. London, Great Britain: Arnold, 1999.
- [3] G. K. von Schulthess, Ed., *Clinical positron emission tomography: correlation with morphological cross-sectional imaging*. Philadelphia, PA: Lippincott Williams and Wilkins, 2000.
- [4] J. M. Ollinger and J. A. Fessler, "Positron-emission tomography," *IEEE Signal Processing Mag.*, vol. 14, pp. 43–55, Jan. 1997.
- [5] S. R. Cherry, J. A. Sorenson, and M. E. Phelps, *Physics in nuclear medicine*, 3rd ed. Philadelphia, PA: Saunders, 2003.
- [6] Z. Liang, "Detector response restoration in image reconstruction of high resolution positron emission tomography," *IEEE Trans. Med. Imag.*, vol. 13, pp. 314–312, June 1994.
- [7] K. Choi, *Mathematical modeling and correction of detector penetration in positron emission tomography*. Master's thesis, Department of Electrical and Computer Engineering, University of Florida, 2002.
- [8] H. Erdoğan and J. A. Fessler, "Monotonic algorithms for transmission tomography," *IEEE Trans. Med. Imag.*, vol. 18, pp. 801–814, Sept. 1999.
- [9] J. M. M. Anderson, R. Srinivasan, B. A. Mair, and J. R. Votaw, "Hidden Markov model based attenuation correction for positron emission tomography," *IEEE Trans. Nucl. Sci.*, vol. 49, pp. 2103–2111, Oct. 2002.
- [10] H. Erdoğan and J. A. Fessler, "Ordered subsets algorithms for transmission tomography," *Physics in Medicine and Biology*, vol. 44, pp. 2835–2851, May 1999.
- [11] B. Bendriem and D. W. Townsend, Eds., *The theory and practice of 3D PET*. Upper Saddle River, NJ: Kluwer Academic Publishers, 1998.
- [12] K. Wienhard, L. Eriksson, S. Grootoorn, M. Casey, U. Pietrzik, and W.-D. Heiss, "Performance evaluation of the positron scanner ECAT EXACT," *Journal of Computer Assisted Tomography*, vol. 16, pp. 804–813, Sept./Oct. 1992.

- [13] J. M. Ollinger, "Detector efficiency and Compton scatter in fully 3D PET," *IEEE Trans. Nucl. Sci.*, vol. 42, pp. 1168–1173, Aug. 1995.
- [14] D. L. Bailey, D. W. Townsend, P. E. Kinahan, S. Grootoont, and T. Jones, "An investigation of factors affecting detector and geometric correction in normalization of 3-D PET data," *IEEE Trans. Nucl. Sci.*, vol. 43, pp. 3300–3307, Dec. 1996.
- [15] W.-H. Lee, J. M. M. Anderson, and J. R. Votaw, "Maximum likelihood estimation of detector efficiencies in positron emission tomography," in *Proc. IEEE Nuclear Science Symposium Conference*, 2001, pp. 2049–2053.
- [16] L. A. Shepp and Y. Vardi, "Maximum likelihood reconstruction for emission tomography," *IEEE Trans. Med. Imag.*, vol. 1, pp. 113–122, Oct. 1982.
- [17] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
- [18] J.-H. Chang, J. M. M. Anderson, and J. R. Votaw, "Regularized image reconstruction algorithms for positron emission tomography," *IEEE Trans. Med. Imag.*, vol. 23, Sept. 2004 (to be published).
- [19] J.-H. Chang, J. M. M. Anderson, and B. A. Mair, "Accelerated penalized maximum likelihood algorithm for positron emission tomography," *IEEE Trans. Image Processing*, under review.
- [20] D. L. Snyder and M. I. Miller, "The use of sieves to stabilize images produced with the EM algorithm for emission tomography," *IEEE Trans. Nucl. Sci.*, vol. 32, pp. 3864–3872, Oct. 1985.
- [21] E. Levitan and G. T. Herman, "A maximum a posteriori probability expectation maximization algorithm for image reconstruction in emission tomography," *IEEE Trans. Med. Imag.*, vol. 6, pp. 185–192, Sept. 1987.
- [22] T. Hebert and R. Leahy, "A generalized EM algorithm for 3-D Bayesian reconstruction from poisson data using Gibbs priors," *IEEE Trans. Med. Imag.*, vol. 8, pp. 194–202, June 1989.
- [23] P. J. Green, "Bayesian reconstructions from emission tomography data using a modified EM algorithm," *IEEE Trans. Med. Imag.*, vol. 9, pp. 84–93, Mar. 1990.
- [24] K. Lange, "Convergence of EM image reconstruction algorithms with Gibbs smoothing," *IEEE Trans. Med. Imag.*, vol. 9, pp. 439–446, Dec. 1990.
- [25] A. R. De Pierro, "On the relation between the ISRA and the EM algorithm for positron emission tomography," *IEEE Trans. Med. Imag.*, vol. 12, pp. 328–333, June 1993.

- [26] C. L. Byrne, "Iterative image reconstruction algorithms based on cross-entropy minimization," *IEEE Trans. Image Processing*, vol. 2, pp. 96–103, Jan. 1993.
- [27] Z. Wu, "MAP image reconstruction using wavelet decomposition," in *Proc. 13th Conference on Information Processing in Medical Imaging*, 1993, pp. 354–371.
- [28] E. Ü. Mumcuoğlu, R. M. Leahy, S. R. Cherry, and Z. Zhou, "Fast gradient-based methods for Bayesian reconstruction of transmission and emission PET images," *IEEE Trans. Med. Imag.*, vol. 13, pp. 687–701, Dec. 1994.
- [29] J. A. Fessler and A. O. Hero, "Penalized maximum-likelihood image reconstruction using space-alternating generalized EM algorithms," *IEEE Trans. Image Processing*, vol. 4, pp. 1417–1429, Oct. 1995.
- [30] A. R. De Pierro, "A modified expectation maximization algorithm for penalized likelihood estimation in emission tomography," *IEEE Trans. Med. Imag.*, vol. 14, pp. 132–137, Mar. 1995.
- [31] C. A. Bouman and K. Sauer, "A unified approach to statistical tomography using coordinate descent optimization," *IEEE Trans. Image Processing*, vol. 5, pp. 480–492, Mar. 1996.
- [32] S. Alenius, U. Ruotsalainen, and J. Astola, "Using local median as the location of the prior distribution in iterative emission tomography image reconstruction," *IEEE Trans. Nucl. Sci.*, vol. 45, pp. 3097–3104, Dec. 1998.
- [33] A. R. De Pierro and M. E. B. Yamagishi, "Fast EM-like methods for maximum a posteriori estimates in emission tomography," *IEEE Trans. Med. Imag.*, vol. 20, pp. 280–288, Apr. 2001.
- [34] I.-T. Hsiao, A. Rangarajan, and G. Gindi, "A new convergent MAP reconstruction algorithm for emission tomography using ordered subsets and separable surrogates," in *Proc. IEEE International Symposium on Biomedical Imaging*, 2002, pp. 409–412.
- [35] S. Ahn and J. A. Fessler, "Globally convergent image reconstruction for emission tomography using relaxed ordered subsets algorithms," *IEEE Trans. Med. Imag.*, vol. 22, pp. 613–626, May 2003.
- [36] B. W. Silverman, M. C. Jones, J. D. Wilson, and D. W. Nychka, "A soothed EM approach to indirect estimation problems, with particular reference to stereology and emission tomography (with discussion)," *Journal of the Royal Statistical Society B*, vol. 52, pp. 271–324, 1990.
- [37] H. Lu and J. M. M. Anderson, "Image reconstruction of PET images using denoised data," in *Proc. IEEE Nuclear Science Symposium Conference*, 2001, pp. 1746–1749.

- [38] H. Lu, Y. Kim, and J. M. M. Anderson, "Improved Poisson intensity estimation: denoising application using Poisson data," *IEEE Trans. Image Processing*, vol. 13, pp. 1128–1135, Aug. 2004.
- [39] W. I. Zangwill, *Nonlinear programming*. Englewood Cliffs, NJ: Prentice-Hall, 1969.
- [40] I.-T. Hsiao, A. Rangarajan, and G. Gindi, "A new convex edge-preserving median prior with applications to tomography," *IEEE Trans. Med. Imag.*, vol. 22, pp. 580–585, May 2003.
- [41] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*, 2nd ed. New York, NY: John Wiley and Sons, 1993.
- [42] J. A. Fessler, "Penalized weighted least-squares image reconstruction for positron emission tomography," *IEEE Trans. Med. Imag.*, vol. 13, pp. 290–300, June 1994.
- [43] D. G. Luenberger, *Linear and nonlinear programming*, 2nd ed. Reading, MA: Addison-Wesley, 1984.
- [44] P. J. Huber, *Robust statistics*. New York, NY: John Wiley and Sons, 1981.
- [45] H. M. Hudson and R. S. Larkin, "Accelerated image reconstruction using ordered subsets of projection data," *IEEE Trans. Med. Imag.*, vol. 13, pp. 601–609, Dec. 1994.
- [46] S. Grootoonk, T. J. Spinks, T. Jones, C. Michel, and A. Bol, "Correction for scatter using a dual energy window technique with a tomograph operated without septa," in *Proc. IEEE Nuclear Science Symposium Conference*, 1991, pp. 1569–1573.
- [47] L. Shao, J. S. Karp, and R. Freifelder, "Composite dual window scattering correction technique in PET," in *Proc. IEEE Nuclear Science Symposium Conference*, 1993, pp. 1391–1395.
- [48] D. W. Townsend, A. Geissbuhler, M. Defrise, E. J. Hoffman, T. J. Spinks, D. L. Bailey, M.-C. Gilardi, and T. Jones, "Fully three-dimensional reconstruction for a PET camera with retractable septa," *IEEE Trans. Med. Imag.*, vol. 10, pp. 505–512, Dec. 1991.
- [49] L. Shao and J. S. Karp, "Cross-plane scattering correction-point source deconvolution in PET," *IEEE Trans. Med. Imag.*, vol. 10, pp. 234–239, Sept. 1991.
- [50] B. T. A. McKee, A. T. Gurvey, P. J. Harvey, and D. C. Howse, "A deconvolution scatter correction for a 3-D PET system," *IEEE Trans. Med. Imag.*, vol. 11, pp. 560–569, Dec. 1992.

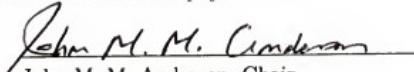
- [51] M. E. Daube-Witherspoon, R. E. Carson, Y. Yan, and T. K. Yap, "Scatter correction in maximum likelihood reconstruction of PET data," in *Proc. IEEE Nuclear Science Symposium and Medical Imaging Conference*, 1992, pp. 945–947.
- [52] J. M. Ollinger, "Model-based scatter correction for fully 3D PET," *Physics in Medicine and Biology*, vol. 41, pp. 153–176, Jan. 1996.
- [53] E. Ü. Mumcuoglu, R. M. Leahy, and S. R. Cherry, "Bayesian reconstruction of PET images: methodology and performance analysis," *Physics in Medicine and Biology*, vol. 41, pp. 1777–1807, Sept. 1996.
- [54] K. Lange, D. R. Hunter, and I. Yang, "Optimization transfer using surrogate objective functions," *Journal of Computational and Graphical Statistics*, vol. 9, pp. 1–20, Mar. 2000.
- [55] T. K. Moon and W. C. Stirling, *Mathematical methods and algorithms*. Upper Saddle River, NJ: Prentice-Hall, 2000.
- [56] K. Lange and R. Carson, "EM reconstruction algorithms for emission and transmission tomography," *Journal of Computer Assisted Tomography*, vol. 8, pp. 306–316, Apr. 1984.
- [57] N. L. Carothers, *Real analysis*. New York, NY: Cambridge University Press, 2000.
- [58] R. G. Bartle, *The elements of real analysis*, 2nd ed. New York, NY: John Wiley and Sons, 1976.
- [59] D. P. Bertsekas, A. Nedić, and A. E. Ozdaglar, *Convex analysis and optimization*. Belmont, MA: Athena Scientific, 2003.
- [60] A. M. Ostrowski, *Solution of equations in Euclidean and Banach spaces*. New York, NY: Academic Press, 1973.
- [61] J. Browne and A. R. De Pierro, "A row-action alternative to the EM algorithm for maximizing likelihoods in emission tomography," *IEEE Trans. Med. Imag.*, vol. 15, pp. 687–699, Oct. 1996.
- [62] J. M. M. Anderson, R. Srinivasan, B. A. Mair, and J. R. Votaw, "Penalized weighted least-squares and maximum likelihood reconstruction of transmission images from PET transmission," *IEEE Trans. Med. Imag.*, under review.
- [63] G. Strang, *Linear algebra and its applications*, 3rd ed. Orlando, FL: Harcourt, 1988.
- [64] A. J. Reader, S. Ally, F. Bakatselos, R. Manavaki, R. J. Walledge, A. P. Jeavons, P. J. Julyan, S. Zhao, D. L. Hastings, and J. Zweit, "One-pass list-mode EM algorithm for high-resolution 3-D PET image reconstruction into large arrays," *IEEE Trans. Nucl. Sci.*, vol. 49, pp. 693–699, June 2002.

- [65] A. J. Reader, P. J. Julyan, H. Williams, D. L. Hastings, and J. Zweit, "EM algorithm system modeling by image-space techniques for PET reconstruction," *IEEE Trans. Nucl. Sci.*, vol. 50, pp. 1392–1397, Oct. 2003.
- [66] S. Kullback and R. Leibler, "On information and sufficiency," *Annals Mathematical Statistics*, vol. 22, pp. 79–86, 1951.
- [67] I. Csiszár and G. Tusnády, "Information geometry and alternating minimization procedures," *Statistics & Decisions*, vol. SI-1, pp. 205–237, 1984.
- [68] R. M. Kessler, J. R. Ellis, Jr., and M. Eden, "Analysis of emission tomographic scan data: limitations imposed by resolution and background," *Journal of Computer Assisted Tomography*, vol. 8, pp. 514–522, 1984.
- [69] G. T. Herman and L. B. Meyer, "Algebraic reconstruction techniques can be made computationally efficient," *IEEE Trans. Med. Imag.*, vol. 12, pp. 600–609, 1993.
- [70] K. Rajan, L. M. Patnaik, and J. Ramakrishna, "Linear array implementation of the EM algorithm for PET image reconstruction," *IEEE Trans. Nucl. Sci.*, vol. 42, pp. 1439–1444, Aug. 1995.
- [71] D. W. Shattuck, J. Rapela, E. Asma, A. Chatzioannou, J. Qi, and R. M. Leahy, "Internet2-based 3D PET image reconstruction using a PC cluster," *Physics in Medicine and Biology*, vol. 47, pp. 2785–2795, July 2002.

BIOGRAPHICAL SKETCH

Ji-Ho Chang was born in Daegu, Korea. He received the B.S. and M.E. degrees in electronic engineering from the Kangwon National University, Chuncheon, Korea, in 1997 and 1999, respectively. He received the M.S. degree in electrical and computer engineering from the University of Florida, Gainesville, Florida, in 2001. For the Ph.D degree, he has been working on developing image reconstruction algorithms for positron emission tomography.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

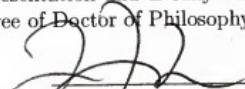

John M. M. Anderson

John M. M. Anderson, Chair
Associate Professor of Electrical and
Computer Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.


Jian Li
Professor of Electrical and Computer
Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.


Fred J. Taylor
Professor of Computer and Information
Science and Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.


Bernard A. Mair
Professor of Mathematics

This dissertation was submitted to the Graduate Faculty of the College of Engineering and to the Graduate School and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

August 2004

Pramod Khargonekar

Pramod P. Khargonekar
Dean, College of Engineering

Kenneth J. Gerhardt
Interim Dean, Graduate School